



Facultad de Ingeniería

Pasantía

Evaluación de Arquitecturas de Software

Architecture Tradeoff Analysis Method *ATAM*



Responsable	
Andrea Delgado	
Integrantes	
Alberto Castro	3.239.979-8
Martín Germán	3.217.050-4

ÍNDICE

1	INTRODUCCIÓN	2
1.1	¿Qué es la Arquitectura de Software?	2
2	EVALUANDO UNA ARQUITECTURA DE SOFTWARE	3
2.1	¿Qué es arquitectónico?	3
2.2	¿Por qué evaluar una Arquitectura?	4
2.3	¿Cuándo una Arquitectura puede ser evaluada?	5
2.3.1	Evaluación temprana	5
2.3.2	Evaluación tardía	5
2.4	¿Quiénes están involucrados?	5
2.4.1	Equipo de evaluación	5
2.4.2	Stakeholders	5
2.5	¿Qué resultado produce la evaluación de una Arquitectura?	6
2.6	¿Por qué cualidades puede ser evaluada una Arquitectura?	6
2.7	¿Por qué los atributos de calidad son demasiados imprecisos para el análisis?	7
2.8	¿Cuáles son las salidas de una evaluación arquitectónica?	8
2.8.1	Lista priorizada de los atributos de calidad requeridos	8
2.8.2	Mapeo de las propuestas a los atributos de calidad	8
2.8.3	Riesgos y no riesgos	8
2.8.4	Catálogo de propuestas arquitectónicas usadas	8
2.8.5	Preguntas de análisis de los atributos de calidad específicos y las propuestas	8
2.8.6	Sensitivity points y tradeoff points	8
2.9	¿Cuáles son los costos y beneficios de realizar una evaluación arquitectónica?	9
2.9.1	Pone a los stakeholders en la misma habitación	9
2.9.2	Fuerza una articulación en las metas específicas de calidad	9
2.9.3	Se priorizan las metas conflictivas	9
2.9.4	Fuerza una explicación clara de la arquitectura	9
2.9.5	Mejora la calidad de la documentación de la arquitectura	9
2.9.6	Descubre oportunidades de reuso	10
2.9.7	Resultan mejoras en las arquitecturas	10
3	ATAM – UN MÉTODO PARA EVALUAR ARQUITECTURAS	11
3.1	Resumen de los pasos del ATAM	11
3.1.1	Presentación	11
3.1.2	Investigación y análisis	11
3.1.3	Pruebas	12
3.1.4	Informes	12
3.2	Descripción en detalle de los pasos del ATAM	12
3.2.1	Paso 1: Presentar el ATAM	12
3.2.2	Paso 2: Presentar las pautas del negocio	13
3.2.3	Paso 3: Presentar la arquitectura	13
3.2.4	Paso 4: Identificar las propuestas arquitectónicas	14
3.2.5	Paso 5: Generar el árbol de utilidad de los atributos de calidad	15
3.2.6	Paso 6: Analizar las propuestas arquitectónicas	17
3.2.7	Paso 7: Lluvia de ideas y priorización de escenarios	24
3.2.8	Paso 8: Analizar las propuestas arquitectónicas	29
3.2.9	Paso 9: Presentar los resultados	29
3.3	Las fases del ATAM	30
3.3.1	Fase 0	31
3.3.2	Fase 1	34
3.3.3	Fase 2	35
3.3.4	Fase 3	37
4	CONCLUSIONES	39
4.1	¿Qué hemos visto de ATAM?	39
4.2	¿Por qué evaluar arquitecturas?	39
4.3	¿Por qué ATAM funciona?	39
5	REFERENCIAS	41

1 INTRODUCCIÓN

1.1 ¿Qué es la Arquitectura de Software?

La Arquitectura de Software es un área de investigación y práctica dentro de la Ingeniería de Software. En particular, la arquitectura de sistemas grandes ha sido objeto de un interés creciente durante la pasada década.

Para empezar, esta materia no nació en forma espontánea en 1990, año en que el término Arquitectura de Software comenzó a ganar aceptación y que fue elemento de especial atención de parte de la industria y la comunidad científica. Este campo fue creado por necesidad, la realidad marcaba que los sistemas de software estaban creciendo, sistemas de cientos de miles de líneas, o incluso de millones de líneas se estaban volviendo comunes.

Existen tres razones por las cuales la Arquitectura de Software es importante para sistemas de software grandes y complejos:

- *Es un vehículo de comunicación entre los stakeholders (interesados).* La Arquitectura de Software es una representación abstracta del sistema, a la que la mayoría de los stakeholders, sino todos, pueden utilizar como base para crear entendimiento mutuo, formar consensos y comunicarse entre ellos.
- *Es una expresión de las decisiones tempranas del diseño.* La Arquitectura de Software de un sistema es el artefacto que permite en forma temprana establecer prioridades entre los diferentes aspectos a ser analizados, y es el artefacto con más influencia en la calidad del sistema. Los aspectos de calidad tales como performance, security, maintainability, reliability, costo del esfuerzo del desarrollo actual y costo del esfuerzo del desarrollo futuro, están todos presentes en la arquitectura.
- *Es una abstracción del sistema reusable y transferible.* La Arquitectura de Software constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo el sistema está estructurado, y como sus componentes trabajan juntos. Este modelo es transferible entre sistemas. En particular, puede ser utilizado en otros sistemas con requerimientos similares, y puede promover reuso a gran escala y en una línea de productos de software.

Hemos comentado sobre el surgimiento e importancia de la Arquitectura de Software, ahora intentaremos definirla:

La Arquitectura de Software de un programa o sistema de computación es la estructura o las estructuras del sistema, que contienen componentes de software, las propiedades externamente visibles de dichos componentes y las relaciones entre ellos.

[Bass, 98]

Hay algunas implicancias claves en esta definición:

- La arquitectura es una abstracción de un sistema o sistemas. Esta representa sistemas en términos de componentes abstractos que tienen propiedades externamente visibles y relaciones.
- Como la arquitectura es abstracta, esta elimina la información local, los detalles de componentes privados no son arquitectónicos.
- Los sistemas están compuestos por muchas estructuras (comúnmente llamadas *vistas*). Una vista por sí sola puede representar nada más que una arquitectura trivial. Es más, una arquitectura debería ser descrita por un conjunto de vistas que soporten las necesidades de su análisis y comunicación.

2 EVALUANDO UNA ARQUITECTURA DE SOFTWARE

La pregunta que surge al definir una arquitectura es: ¿cómo puedo estar seguro que la arquitectura elegida es la correcta para mi software?

No es una pregunta fácil de responder, lo que si sabemos es que la arquitectura es la piedra fundamental de cualquier software. La arquitectura es quien define los atributos de calidad de un sistema.

Hasta hace poco, no existían métodos de utilidad para validar arquitecturas, que respondieran con confianza a la pregunta planteada.

Antes de continuar, sigamos analizando la definición de arquitectura tomada:

La Arquitectura de Software de un programa o sistema de computación es la estructura o las estructuras del sistema, que contienen componentes de software, las propiedades externamente visibles de dichos componentes y las relaciones entre ellos.

[Bass, 98]

Por propiedades "externamente visibles" nos estamos refiriendo a lo que otros componentes asumen de lo que hace un componente, como ser los servicios que provee, características de performance, manejo de fallas, entre otras.

La arquitectura define componentes (como ser módulos, objetos, procesos, subsistemas, entre otros) y relaciones relevantes (como llamadas, sincronismos, dependencias, entre otras). La arquitectura es el resultado de las decisiones tempranas de diseño, que son necesarias antes de empezar a construir el sistema.

Para poder entender claramente que es evaluar una arquitectura, hay que tener claro previamente el siguiente concepto:

La Arquitectura permite o excluye prácticamente todos los atributos de calidad del sistema.

Esto nos lleva a una verdad fundamental sobre la evaluación de arquitecturas:

Si las decisiones arquitectónicas determinan los atributos de calidad del sistema, entonces es posible evaluar las decisiones arquitectónicas con respecto a su impacto sobre dichos atributos.

2.1 ¿Qué es arquitectónico?

Siempre en algún momento surge la pregunta: "¿Qué es arquitectónico?" De repente no surge exactamente de esta manera, pero puede ser preguntada de las siguientes formas:

- ¿Cuál es la diferencia entre arquitectura y diseño de alto nivel?
- ¿Son arquitectónicos los detalles como las prioridades de los procesos?
- ¿Por qué debe ser considerados los detalles de implementación como ser los desbordes de buffers?
- ¿Son las interfaces de los componentes parte de la arquitectura?
- Si tengo el diagrama de clases, ¿necesito hacer algo más?
- ¿Está la arquitectura involucrada con el comportamiento en tiempo de ejecución o la estructura estática?
- ¿Es el sistema operativo parte de la arquitectura? ¿Lo es el lenguaje de programación?
- Si estoy condicionado a utilizar un producto comercial (COTS), ¿es arquitectónico? Si puedo elegir cualquier producto comercial, ¿es arquitectónico?

Razonemos de dos maneras:

Consideremos la definición de arquitectura que estamos tomando. Parafraseando: Es de interés para la Arquitectura de Software la organización del sistema a alto nivel, describiéndola en términos de componentes, sus propiedades externamente visibles, y las relaciones entre ellos. Esto último es verdadero, pero falla cuando nos centralizamos en determinados contextos, dado que el contexto influye en lo que es arquitectónico. Por ejemplo, si mi contexto es un subsistema de un sistema dado, lo que es arquitectónico para mí, no lo es para el arquitecto del sistema completo.

Preguntémonos ahora, ¿qué no es arquitectónico? Se ha dicho que los algoritmos no son arquitectónicos, las estructuras de datos no son arquitectónicas, los detalles del flujo de datos no son arquitectónicos. Esto último, no es del todo verdadero. Por ejemplo, algunas propiedades de los algoritmos, como ser su complejidad, pueden tener un fuerte impacto sobre la performance.

Entonces, ¿existe un principio para determinar qué es arquitectónico? Formulémoslo a partir del uso que se le da a una arquitectura. Nuestro criterio para determinar que es arquitectónico sería entonces:

Debe ser un componente, relación entre componentes, o una propiedad (de componentes o relaciones) que necesita ser externamente visible, con el objetivo de razonar sobre la habilidad del sistema de alcanzar sus requerimientos de calidad, o de soportar la descomposición del sistema en partes independientemente implementables.

Aquí presentamos corolarios de este principio:

- *La arquitectura describe que hay en el sistema.* Cuando se determina el contexto, se determinan las fronteras que describen que hay dentro y que hay fuera del sistema. La arquitectura describe lo que hay dentro.
- *Una arquitectura es una representación abstracta del sistema.* La información en una arquitectura es abstracta y sin embargo es la representación más significativa del sistema. Dada la especificación de la arquitectura, no se debería necesitar ninguna otra descripción abstracta.
- *Lo que es arquitectónico debería ser crítico para razonar acerca de requerimientos críticos.* La arquitectura es el nexo entre los requerimientos y el diseño. Si alguna información es crítica para argumentar si el sistema alcanzará los requerimientos de calidad, entonces es arquitectónico.
- *La especificación de una arquitectura necesita ser comprensible.* La idea de una representación del sistema a alto nivel, es que se puede entender y trabajar con ella. Demasiado detalle podría ir en contra de este propósito.
- *Una arquitectura debe restringir.* Esta impone requerimientos a todos los diseños de bajo nivel.

En resumen, ser arquitectónico, es ser la representación más abstracta del sistema, que permite razonar acerca de los requerimientos críticos y las restricciones a todos los niveles de refinamiento subsecuentes.

2.2 ¿Por qué evaluar una Arquitectura?

Cuanto más temprano se encuentre un problema en un proyecto de software, mejor. El costo de arreglar un error durante las fases de requerimientos o diseño, es mucho menor al costo de arreglar ese mismo error en la fase de verificación. Dado que la arquitectura es un producto temprano de la fase de diseño, esta tiene un profundo efecto en el sistema y en el proyecto.

Una mala arquitectura puede llevar a un proyecto al fracaso. Todos los requerimientos de calidad pueden quedar insatisfechos.

La arquitectura también determina la estructura del proyecto: configuración, agenda y presupuesto, alcance, entre otros aspectos. Es mejor cambiar la arquitectura antes que otros artefactos, que están basados en ella, se establezcan.

Realizar una evaluación de la arquitectura es la manera más económica de evitar desastres.

2.3 ¿Cuándo una Arquitectura puede ser evaluada?

Generalmente, la evaluación de la arquitectura ocurre después que esta ha sido especificada, pero antes que empiece la implementación. En un proceso iterativo y/o incremental, la evaluación se puede realizar al final de cada ciclo. Sin embargo, uno de los atractivos de la evaluación de arquitecturas es que se puede efectuar en cualquier etapa de la vida de una arquitectura. En particular, existen dos variaciones útiles: temprana y tardía.

2.3.1 Evaluación temprana

La evaluación no tiene porque esperar a que la arquitectura este totalmente especificada. Esta puede ser utilizada en cualquier etapa del proceso de creación de la arquitectura, para examinar las decisiones arquitectónicas ya tomadas y decidir entre las opciones que están pendientes.

Por supuesto, la completitud y fidelidad de la evaluación es directamente proporcional a la completitud y fidelidad de la descripción de la arquitectura.

2.3.2 Evaluación tardía

Esta variación toma lugar no solo cuando la arquitectura esta terminada, también cuando la implementación esta completa. Este caso ocurre cuando la organización hereda un sistema legado. La técnica para evaluar un sistema legado es la misma que para evaluar un sistema recién nacido. Una evaluación es útil para entender el sistema legado, y saber si este cumple con los requerimientos de calidad y comportamiento.

En general, una evaluación debe realizarse cuando hay suficiente de la arquitectura como para justificarlo. Una buena regla sería: *realizar una evaluación cuando el equipo de desarrollo empieza a tomar decisiones que dependen de la arquitectura y el costo de deshacerlas sobrepasa al costo de realizar una evaluación.*

2.4 ¿Quiénes están involucrados?

Hay dos grupos de personas involucrados en la evaluación de la arquitectura.

2.4.1 Equipo de evaluación

Estas son las personas que conducirán la evaluación y realizaran el análisis.

2.4.2 Stakeholders

Estos son los interesados en la arquitectura, y en el sistema que se construirá a partir de ella. Algunos de los stakeholders, pero no todos, serán miembros del equipo de desarrollo, como ser implementadores, verificadores, entre otros. Un tipo especial de stakeholder es el *decision maker* (tomador de decisiones) del proyecto. Estos son los interesados en la salida de la evaluación y tienen el poder de toma de decisiones. Algunos de ellos son el arquitecto, los diseñadores y el director de proyecto. Cuando un stakeholder quiere que algo sea verdad en la arquitectura, un decision maker tiene el poder de asignar recursos para hacerlo realidad. Generalmente el cliente de la evaluación de la arquitectura es un decision maker, con un gran interés en la salida de la misma y poder sobre todo el proyecto.

En ocasiones el equipo de evaluación esta formado por integrantes del plantel del proyecto, en cuyo caso también son stakeholders. Esto no está recomendado dado que habrá pérdida de la objetividad a la hora de ver la arquitectura.

2.5 ¿Qué resultado produce la evaluación de una Arquitectura?

En términos concretos, la evaluación de la arquitectura produce un informe, la forma y contenido del mismo varía según el método utilizado. En particular, produce repuestas a dos tipos de preguntas:

- ¿Es esta arquitectura adecuada para el sistema para la cual fue diseñada?
- ¿Cuál de dos o más arquitecturas propuestas es la más adecuada para el sistema?

Decimos que una arquitectura es adecuada cuando cumple dos criterios:

- El sistema resultante de ella cumple con los objetivos de calidad. No todas las propiedades de calidad del sistema son resultado directo de la arquitectura, pero muchas lo son.
- El sistema puede ser construido con los recursos con que se cuenta: el plantel, el presupuesto, el sistema legado (si hay), entre otros. Esto es, la arquitectura es construible.

Un resultado que también produce la evaluación de una arquitectura es la captura y priorización de las metas que la arquitectura debe cumplir para poder ser considerada adecuada.

La evaluación de una arquitectura no produce resultados cuantitativos. No es de interés, por ejemplo, evaluar la performance en cantidad de transacciones por segundo a esta altura, dado que el sistema no está construido aún. Lo que interesa, en un espíritu de mitigación de riesgos, es aprender como un atributo de calidad es afectado por una decisión de diseño arquitectónico, para que de esta manera se pueda estudiar con cuidado dicha decisión.

Una evaluación arquitectónica dice si una arquitectura es adecuada respecto a un conjunto de metas, y problemática con respecto a otro conjunto de metas. En ocasiones, las metas pueden ser contradictorias entre ellas, o algunas ser más importantes que otras. El director de proyecto es quien deberá tomar la decisión si la arquitectura evalúa bien o mal en las distintas áreas. La evaluación ayuda a encontrar debilidades, no dirá "sí" o "no", "bien" o "mal", "6 en 10", dirá donde están los riesgos.

2.6 ¿Por qué cualidades puede ser evaluada una Arquitectura?

No es del todo cierto que podamos decir que el sistema alcanzará todas sus metas de calidad con solo mirar la arquitectura. Pero muchos atributos de calidad yacen directamente en el reino de la arquitectura. En particular, el ATAM se concentra en evaluar la adecuación de una arquitectura en base a los siguientes atributos de calidad:

- *Performance*: Refiere a la *responsiveness* del sistema. Es el tiempo requerido para responder a un estímulo (evento), o número de eventos por unidad de tiempo. Por ejemplo, una posible medida podría ser cantidad de transacciones por segundo.
- *Reliability*: Es la habilidad del sistema de continuar operando sobre el tiempo. Es usualmente medida en tiempo promedio entre fallas.
- *Availability*: Es la porción de tiempo en que el sistema está levantado y corriendo. Se mide como el tiempo transcurrido entre fallas, así como, cuán rápido el sistema está apto para reanudar y quedar operativo ante una falla.
- *Security*: Es la medida de la habilidad del sistema de resistirse al uso no autorizado y negar los servicios, mientras los provee a usuarios legítimos.
- *Modifiability*: Es la habilidad de realizar cambios al sistema en forma rápida y a bajo costo.
- *Portability*: Es la habilidad del sistema de correr sobre diferentes ambientes. Estos ambientes pueden ser de hardware, de software o una combinación de ambos. Portability es un caso particular de modifiability.

- *Functionality*: Es la habilidad del sistema de hacer el trabajo para el cual fue construido.
- *Variability*: Es la capacidad de la arquitectura de ser expandida o modificada para producir nuevas arquitecturas. Variability es importante cuando la arquitectura se va a utilizar como piedra fundamental de toda una familia de productos relacionados, como ser una línea de producto.
- *Subsetability*: Es la habilidad de soportar la producción de un subconjunto del sistema. Subsetability permite el desarrollo incremental. Es un tipo especial de variability.
- *Conceptual integrity*: Es la visión que unifica el diseño del sistema en todos los niveles. La arquitectura debe hacer cosas similares en forma similar. Debe mostrar consistencia en los mecanismos, decisiones y patrones aplicados.

Si otro atributo de calidad, además de los mencionados antes, es importante para determinado proyecto, este también debe formar parte de la evaluación. En particular, el ATAM permite definir nuevos atributos de calidad a ser utilizados en la evaluación.

2.7 ¿Por qué los atributos de calidad son demasiados imprecisos para el análisis?

Los atributos de calidad son la base para a la evaluación de una arquitectura, pero por si solos no son suficiente para juzgar la adecuabilidad de la arquitectura. Generalmente, los requerimientos son escritos de la siguiente manera:

- "El sistema debe ser robusto."
- "El sistema debe ser modificable."
- "El sistema debe ser seguro."
- "El sistema debe tener una performance aceptable."

Cada una de las frases anteriores esta sujeta a diferentes interpretaciones y malos entendidos. Lo que uno puede considerar robusto, otro puede considerarlo apenas aceptable.

El punto es que los atributos de calidad no son cantidades absolutas, existen en un contexto con metas específicas. En particular:

- Un sistema es modificable (o no) con respecto a un tipo de cambio específico.
- Un sistema es seguro (o no) con respecto a un tipo de amenaza específica.
- Un sistema es confiable (o no) con respecto a la ocurrencia de un tipo de falta específica.
- Un sistema es performante (o no) con respecto a un criterio específico de performance.
- Una arquitectura es construible (o no) con respecto a restricciones de tiempo y presupuesto específicas.

No parece razonable, considerar que un sistema pueda alguna vez, por ejemplo, ser completamente confiable bajo toda circunstancia (pensar en problemas de energía, meteorológicos, entre otros). Dado esto, es importante que el arquitecto entienda perfectamente bajo que circunstancias un sistema debe ser confiable para ser considerado aceptable. Por lo tanto, el primer trabajo que debe realizar una evaluación de arquitectura es obtener las metas específicas de calidad ante las cuales la arquitectura será juzgada.

Si algunas de estas metas no son específicas o son ambiguas, se debe pedir a los stakeholders que ayuden al equipo de evaluación a reescribirlas. El mecanismo a utilizar para representar estas metas es el de *escenario*. Un escenario es una pequeña descripción de la interacción de un stakeholder con el sistema. Los escenarios son parecidos a los *casos de uso*.

2.8 ¿Cuáles son las salidas de una evaluación arquitectónica?

Las salidas de una evaluación arquitectónica son información e ideas sobre la arquitectura.

En particular, el ATAM produce las salidas que se describen a continuación.

2.8.1 Lista priorizada de los atributos de calidad requeridos

Una evaluación arquitectónica solo puede proceder si se conoce el criterio de adecuabilidad. Es más, la obtención de los atributos de calidad requeridos contra los cuales la arquitectura será juzgada, constituye la mayor parte del trabajo. Pero ninguna arquitectura puede tener una lista interminable de atributos de calidad, y por lo tanto, se utilizan métodos de priorización consensuados.

2.8.2 Mapeo de las propuestas a los atributos de calidad

Las respuestas a las preguntas de análisis producen un mapeo que muestra como las propuestas arquitectónicas logran (o no) los atributos de calidad deseados. Este mapeo crea una espléndida base lógica para la arquitectura.

2.8.3 Riesgos y no riesgos

Los riesgos son decisiones arquitectónicas potencialmente problemáticas. Los no riesgos son buenas decisiones, que confían en asunciones que con frecuencia son implícitas en la arquitectura.

Documentar riesgos y no riesgos consiste en:

- Una decisión arquitectónica (o una decisión que no ha sido tomada)
- Una respuesta específica al atributo de calidad que esta siendo tratado, junto con las consecuencias del nivel predecible de la respuesta.
- Una base lógica por el efecto positivo o negativo que la decisión tuvo en alcanzar el requerimiento del atributo de calidad.

Para que un no riesgo se mantenga como tal, las asunciones no deben cambiar, o al menos si cambian, la designación como no riesgo debe ser rejustificada.

2.8.4 Catálogo de propuestas arquitectónicas usadas

Todo arquitecto adopta ciertas estrategias y propuestas para resolver los problemas. A veces estas propuestas son conocidas, y a veces no, son innovadoras. En ambos casos, son la clave para entender como la arquitectura alcanzará sus metas y requerimientos. El ATAM crea un catálogo con las propuestas arquitectónicas, para que las personas puedan entender y familiarizarse con la arquitectura.

2.8.5 Preguntas de análisis de los atributos de calidad específicos y las propuestas

El ATAM posee preguntas de análisis que están basadas en los atributos que están siendo buscados y en las propuestas seleccionadas por el arquitecto. Como la arquitectura evoluciona, estas preguntas pueden ser utilizadas en el futuro como mini evaluaciones para estar seguros que la evolución no esta llevando a la arquitectura por un camino equivocado.

2.8.6 Sensitivity points y tradeoff points

Denominamos *sensitivity points* (puntos de sensibilidad) y *tradeoff points* (puntos de intercambio) a las decisiones arquitectónicas claves. Un sensitivity point es una decisión arquitectónica que involucra uno o más componentes arquitectónicos (y/o relaciones de componentes) que es crítico para lograr una respuesta medida de un atributo de calidad particular. Se llama así porque la medida de la respuesta es sensible a los cambios de decisión.

Por ejemplo:

- El nivel de confidencialidad en una red privada virtual puede ser sensible al número de bits de encriptación.
- La latencia por procesar un mensaje importante puede ser sensible a la prioridad del proceso de menor prioridad involucrado en el manejo del mensaje.

Los sensitivity points le dicen al diseñador o analista donde están los focos de atención cuando esta tratando de entender el logro de una meta de calidad. Sirven de bandera amarilla: "Tenga cautela cuando cambie esta propiedad de la arquitectura".

Un tradeoff point es una decisión arquitectónica que afecta a más de un atributo y es un sensitivity point de más de un atributo. Por ejemplo, cambiar el nivel de encriptación puede tener un efecto significativo en la seguridad y la performance. Incrementar el nivel de encriptación puede mejorar la seguridad, pero requiere más tiempo de procesamiento. Los tradeoff points son las decisiones más críticas que uno puede hacer en una arquitectura, por eso el ATAM se centra particularmente en ellos. No olvidar, que es tan importante marcar con banderas las decisiones claves tomadas, como aquellas decisiones claves que aún no han sido tomadas.

2.9 ¿Cuáles son los costos y beneficios de realizar una evaluación arquitectónica?

El mayor beneficio que brinda la evaluación de una arquitectura, es que descubre los problemas que si se hubiesen dejado sin descubrir, habría sido mucho más costoso corregirlos luego. En breve, una evaluación produce una mejor arquitectura.

Algunos beneficios más que brinda la evaluación se presentan a continuación.

2.9.1 Pone a los stakeholders en la misma habitación

En una evaluación arquitectónica es, comúnmente, la primera vez en que la mayoría de los stakeholders se encuentran. Es más, es la primera vez que el arquitecto se encuentra con ellos. Todos comparten un objetivo, lograr un sistema exitoso.

2.9.2 Fuerza una articulación en las metas específicas de calidad

El rol del stakeholder es articular las metas de calidad que la arquitectura debería alcanzar para ser considerada exitosa. Estas metas no siempre son capturadas en algún documento de requerimientos.

2.9.3 Se priorizan las metas conflictivas

Los conflictos que pueden aparecer entre las metas expresadas por los distintos stakeholders serán ventiladas. Si el arquitecto no puede satisfacer a todas las metas en conflicto, se le dirá cuales son más importantes.

2.9.4 Fuerza una explicación clara de la arquitectura

El arquitecto convoca a un grupo de personas, no en privado, para explicarles la creación de la arquitectura, en detalle y sin ambigüedades. El proyecto se ve beneficiado cuanto antes se realice esta explicación.

2.9.5 Mejora la calidad de la documentación de la arquitectura

Generalmente, la evaluación pide documentación que aún no esta terminada, entonces se designa alguien del plantel a terminarla. Una vez más, el proyecto se ve beneficiado porque se ingresa al desarrollo mejor preparado al tener los documentos terminados.

2.9.6 Descubre oportunidades de reuso

Los stakeholders y el equipo de evaluación provienen de afuera del proyecto de desarrollo, pero trabajan o están familiarizados con otros proyectos. Por lo tanto, ambos están en una buena posición para detectar componentes que pueden ser reusados en otros proyectos, o conocer componentes que ya existen que pueden ser utilizados en el proyecto actual.

2.9.7 Resultan mejoras en las arquitecturas

Las organizaciones que practican la evaluación arquitectónica como un estándar de su proceso de desarrollo, reportan una mejora en la calidad de las arquitecturas que son evaluadas. La evaluación de arquitecturas resulta en mejores arquitecturas no solo después de hecha, pero antes también.

Los costos de la evaluación arquitectónica son todos costos de personal y costos de oportunidad por aquel personal que participa de la evaluación en vez de hacer otra cosa. Estos costos son sencillos de calcular (están tabulados).

3 ATAM – UN MÉTODO PARA EVALUAR ARQUITECTURAS

El Architecture Tradeoff Analysis Method (ATAM) obtiene su nombre no solo porque nos dice cuan bien una arquitectura particular satisface las metas de calidad, sino que también provee ideas de cómo esas metas de calidad interactúan entre ellas, como realizan concesiones mutuas entre ellas.

Tener un método estructurado, permite hacer el análisis repetible y ayuda a asegurar que las preguntas acerca de una arquitectura serán contestadas en forma temprana, cuando es relativamente económico corregir problemas.

El ATAM también puede ser utilizado para analizar sistemas legados. Esta necesidad nace cuando el sistema legado necesita ser modificado, integrado con otro sistema, entre otras necesidades. Aplicar el ATAM incrementa el entendimiento de los atributos de calidad del sistema legado.

3.1 Resumen de los pasos del ATAM

La parte principal del ATAM consiste de nueve pasos. Estos pasos se dividen cuatro grupos:

- *Presentación*, donde la información es intercambiada.
- *Investigación y análisis*, donde se valoran los atributos claves de calidad requeridos, uno a uno con las propuestas arquitectónicas.
- *Pruebas*, donde se revisan los resultados obtenidos contra las necesidades relevantes de los stakeholders.
- *Informes*, donde se presentan los resultados del ATAM.

3.1.1 Presentación

1. *Presentar el ATAM*

El líder del equipo de evaluación describe el método a los participantes, fija las expectativas y responde las preguntas que puedan surgir.

2. *Presentar las pautas del negocio*

Un representante del proyecto (por ejemplo, el director de proyecto o el cliente del sistema) describe las metas del negocio que motivan el esfuerzo de desarrollo y que se convertirán en las pautas primordiales de la arquitectura (por ejemplo, alta disponibilidad o alta seguridad).

3. *Presentar la arquitectura*

El arquitecto describe la arquitectura, enfocándose en como esta sigue las pautas del negocio.

3.1.2 Investigación y análisis

4. *Identificar las propuestas arquitectónicas*

Las propuestas arquitectónicas son identificadas por el arquitecto, pero no son analizadas.

5. Generar el árbol de utilidad de los atributos de calidad

Los atributos de calidad que comprometen la utilidad del sistema (performance, availability, entre otros) son obtenidos, especificados en escenarios (con estímulos y respuestas) y priorizados.

6. Analizar las propuestas arquitectónicas

Basados en los escenarios de mayor prioridad identificados en el paso 5, las propuestas arquitectónicas que cumplen con estos escenarios, son obtenidas y analizadas (por ejemplo, una propuesta arquitectónica que logra una meta de performance, será objeto de un análisis de performance).

Durante este paso los riesgos arquitectónicos, los no riesgos, los sensitivity points y tradeoff points son identificados.

3.1.3 Pruebas

7. Lluvia de ideas y priorización de escenarios

Un gran conjunto de escenarios es obtenido por todos los stakeholders. Este conjunto es priorizado mediante votación.

8. Analizar las propuestas arquitectónicas

En este paso se reitera las actividades del paso 6 utilizando el ranking de escenarios del paso 7. Estos escenarios se consideran casos de prueba para confirmar el análisis realizado hasta ahora. Este análisis puede descubrir nuevas propuestas arquitectónicas, riesgos, no riesgos, sensitivity points y tradeoff points, que son documentados.

3.1.4 Informes

9. Presentar los resultados

Basados en la información recogida durante el ATAM (propuestas, escenarios, árbol de utilidad, riesgos, no riesgos, sensitivity points y tradeoff points), el equipo de ATAM presenta los resultados a los stakeholders.

A veces, se deben hacer modificaciones dinámicas al calendario para poder acomodarse a la disponibilidad del personal o de la información arquitectónica. A pesar que los pasos están numerados, sugiriendo linealidad, este no es un proceso en cascada estricto. De vez en cuando, un evaluador podrá volver a pasos anteriores, saltar a un paso posterior, o iterar entre pasos, según la necesidad. La importancia de los pasos esta claramente delineada en las actividades involucradas en el ATAM, junto con las salidas de estas actividades.

3.2 Descripción en detalle de los pasos del ATAM

3.2.1 Paso 1: Presentar el ATAM

En el primer paso el líder del equipo de evaluación presenta el ATAM a los stakeholders. Este tiempo es utilizado para explicar el proceso que todos seguirán, responder preguntas, y fijar el contexto y las expectativas para el resto de las actividades. En particular, el líder describe:

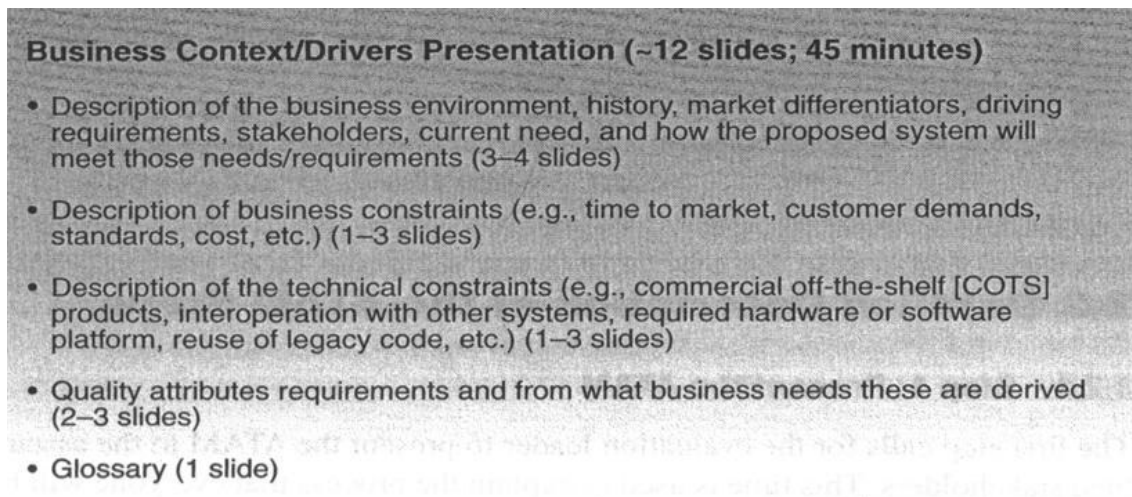
- Los pasos del ATAM en resumen.
- Las técnicas que serán utilizadas para la obtención y análisis: generación del árbol de utilidad, obtención y análisis de las propuestas arquitectónicas, lluvia de ideas y priorización de escenarios.

- Las salidas de la evaluación: los escenarios obtenidos y priorizados, las preguntas utilizadas para entender y evaluar la arquitectura, el árbol de utilidad describiendo y priorizando los requerimientos que guían la arquitectura, un conjunto de propuestas arquitectónicas identificadas, un conjunto de riesgos y no riesgos descubiertos, y un conjunto de sensitivity points y tradeoff points descubiertos.

3.2.2 Paso 2: Presentar las pautas del negocio

Los participantes de la evaluación, los stakeholders así como el equipo de evaluación, necesitan entender el contexto del sistema y las principales pautas del negocio que motivan el desarrollo. En este paso, un decision maker del proyecto (por ejemplo, el director de proyecto o el cliente del sistema) presenta el sistema desde la perspectiva del negocio. La presentación debería describir:

- Las funciones más importantes del sistema.
- Toda restricción técnica, dirigenal, económica o política relevante.
- Como las metas del negocio y el contexto se relacionan con el proyecto.
- La mayoría de los stakeholders.
- Las guías de la arquitectura (mayoritariamente las metas de los atributos de calidad que dan forma a la arquitectura).



Plantilla de ejemplo para presentación del negocio.

3.2.3 Paso 3: Presentar la arquitectura

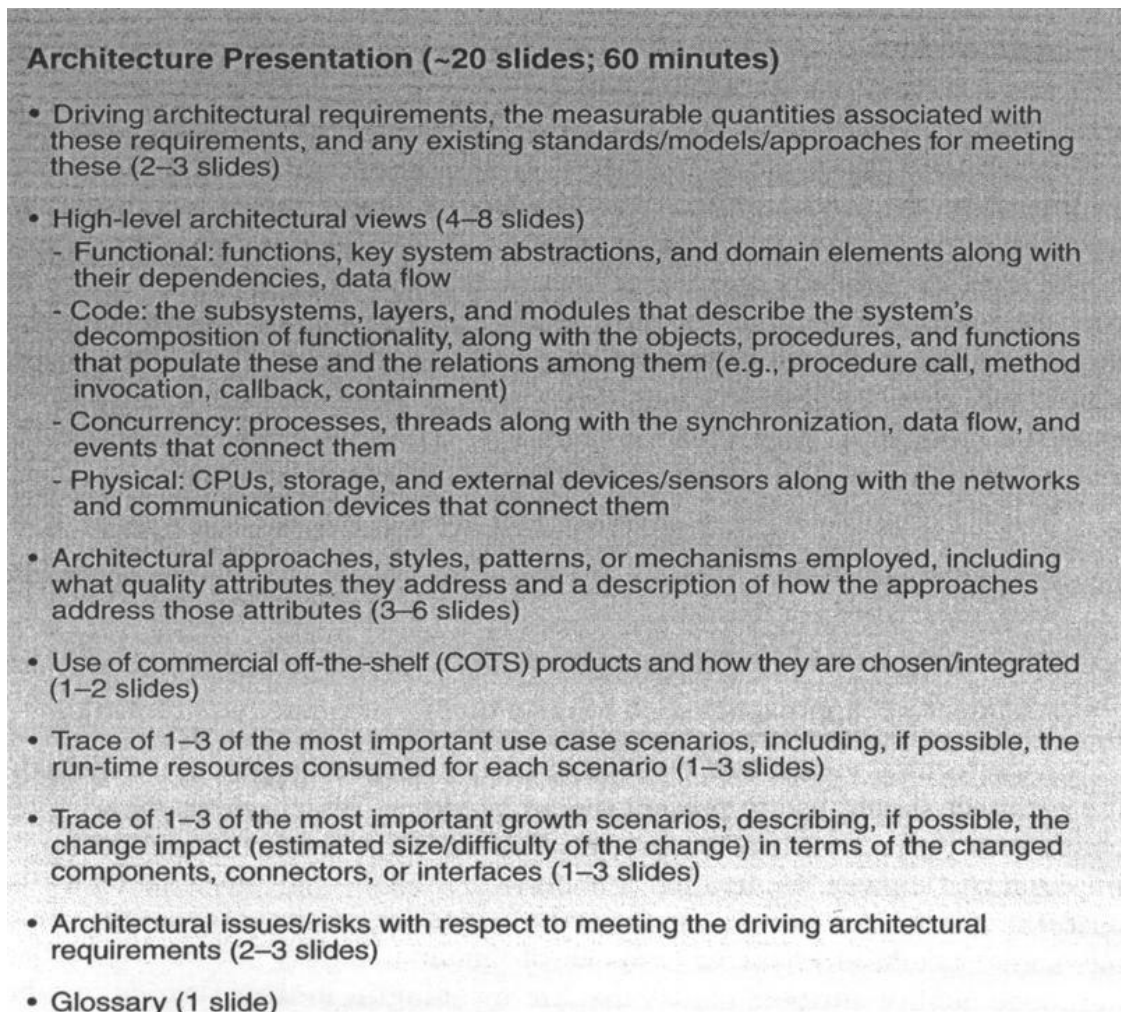
En este paso, el arquitecto líder (o equipo de arquitectura) realiza una presentación describiendo la arquitectura en un nivel apropiado de detalle. Cual es el "nivel apropiado" depende de muchos factores: cuanto de la arquitectura ha sido diseñada y documentada, cuanto tiempo hay disponible, y de los requerimientos de calidad. La información arquitectónica presentada afectará directamente el posible análisis y la calidad del mismo.

La presentación de la arquitectura debería cubrir:

- Las restricciones técnicas, como el sistema operativo, el hardware, o middleware prescrito para el uso.
- Otros sistemas con los cuales el sistema debe interactuar.
- Propuestas arquitectónicas utilizadas para alcanzar los requerimientos de los atributos de calidad.

Las vistas contenidas en el documento de la arquitectura, son el principal vehículo que el arquitecto debería utilizar para presentar la arquitectura.

En este momento el equipo de evaluación comienza su prueba inicial capturando las propuestas arquitectónicas como preludeo del paso 4.



Plantilla de ejemplo para presentación de la arquitectura.

3.2.4 Paso 4: Identificar las propuestas arquitectónicas

El ATAM centraliza el análisis de una arquitectura en el entendimiento de sus propuestas arquitectónicas. En este paso, son capturadas por el equipo de evaluación pero no son analizadas. El equipo le pedirá al arquitecto que explícitamente nombre toda propuesta usada, pero ellos también capturarán todas las propuestas que hayan escuchado durante la presentación del arquitecto en el paso previo.

El ATAM se concentra en identificar propuestas arquitectónicas y estilos arquitectónicos porque estos representan la manera en que la arquitectura cumple con los atributos de calidad de mayor prioridad, esto significa que se asegura que los requerimientos críticos serán alcanzados de manera predecible. Estas propuestas arquitectónicas definen las estructuras importantes del sistema, y describen como el sistema puede crecer, responder a cambios, entre otros.

Un estilo arquitectónico incluye una descripción de los tipos de componentes y su topología, una descripción del patrón de datos y control de interacciones entre componentes, y una descripción informal de los beneficios y perjuicios que tiene.

Un estilo puede ser considerado como un conjunto de restricciones sobre una arquitectura, restricciones sobre los tipos de componentes y sus interacciones, y como esas restricciones definen una familia de arquitecturas que las satisfacen. Al descubrir estilos arquitectónicos en la arquitectura, los evaluadores pueden ver que estrategias utilizó el arquitecto para responder a las metas establecidas por los atributos de calidad del sistema.

3.2.5 Paso 5: Generar el árbol de utilidad de los atributos de calidad

En este paso el equipo de evaluación trabaja junto con los decision makers (el equipo de arquitectura, el director y los clientes representativos) para identificar, priorizar y refinar las metas de los atributos de calidad más importantes del sistema. Este paso crucial guía el resto del análisis. Sin esta guía, los evaluadores pueden perder su tiempo analizando aspectos de la arquitectura que no son de interés para los stakeholders. Debe haber una manera en la cual los evaluadores y los stakeholders concentren su atención en los aspectos de la arquitectura que son más críticos para el éxito del sistema. Esto es conseguido construyendo un árbol de utilidad.

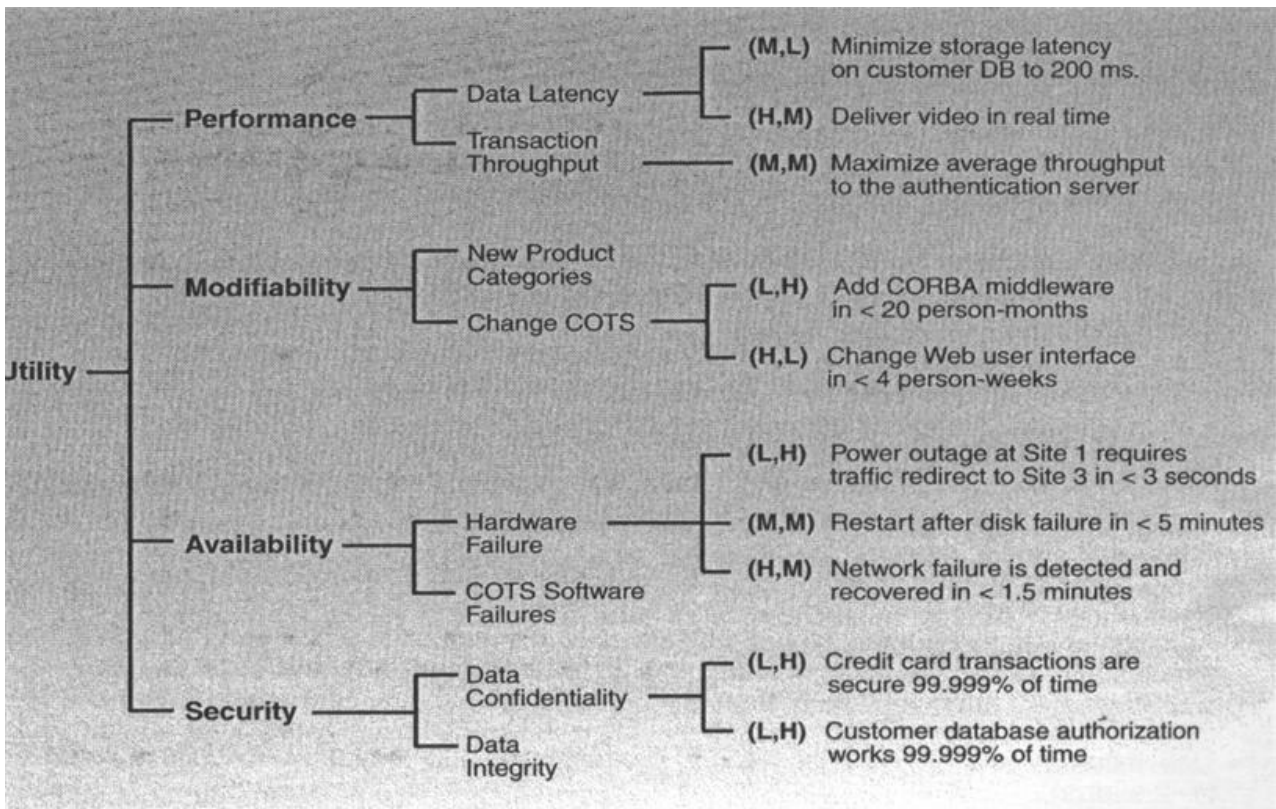
La salida de la generación del árbol de utilidad es una lista priorizada de los requerimientos de los atributos de calidad, comprendidos como escenarios. Le dice al equipo de evaluación donde gastar su tiempo, y en particular donde buscar propuestas arquitectónicas y riesgos. El árbol de utilidad guía a los evaluadores a buscar propuestas arquitectónicas que estén involucradas en la satisfacción de los escenarios de alta prioridad que se encuentran en las hojas del árbol de utilidad. Es más, el árbol de utilidad sirve para hacer que los requerimientos de los atributos de calidad sean concretos, forzando al arquitecto y a los clientes representativos a que definan en forma precisa los requerimientos de calidad relevantes.

El árbol de utilidad provee un mecanismo que en forma directa y eficiente traduce las pautas del negocio del sistema en escenarios concretos de los atributos de calidad.

Los participantes priorizan al árbol de utilidad en dos dimensiones:

1. Por la importancia que cada escenario tiene para el éxito del sistema.
2. Por el grado de dificultad que posee el escenario para ser realizado, según la estimación del arquitecto.

Habitualmente la escala utilizada para ambas dimensiones es **High, Medium, Low**.



Ejemplo de árbol de utilidad.

3.2.5.1 Escenarios

Típicamente el primer trabajo que se hace en una evaluación es precisamente obtener las metas de calidad específicas contra las cuales la arquitectura será juzgada. El mecanismo que usa para esta obtención es el de escenario.

Un escenario es una corta descripción de una interacción de un stakeholder con el sistema. Por ejemplo, un *usuario* describe como utiliza el sistema para realizar una tarea, sus escenarios serán muy parecidos a los *casos de uso*. Un escenario de un desarrollador se centrará en usar la arquitectura para construir el sistema o predecir su performance.

Los escenarios proveen un vehículo para tomar las cualidades de *tiempo de desarrollo* vagas, como ser modifiability, y transformarlas en algo concreto: ejemplos específicos sobre el uso actual y futuro del sistema. Los escenarios también son útiles para entender las cualidades de *tiempo de ejecución* como performance o availability.

Estructura de los escenarios:

Los escenarios cuentan una pequeña historia sobre una interacción con el sistema desde el punto de vista de un stakeholder. Se le pide a los stakeholders en un ejercicio de evaluación que escriban escenarios utilizando el formato tripartito. Las tres partes son *estímulo*, *entorno* y *respuesta*.

El *estímulo* es la parte del escenario que explica o describe lo que hace un stakeholder para iniciar una interacción con el sistema, por ejemplo invocar una función.

El *entorno* describe que esta pasando al tiempo del estímulo. Por ejemplo, ¿cuál es el estado del sistema?, ¿cuáles son las condiciones inusuales? Cualquier condición del

entorno que es relevante para entender el escenario debería ser descrita. Por convención, si el entorno es "en condiciones normales", entonces se omite.

La *respuesta* dice como el sistema, a través de la arquitectura, debería responder al estímulo. Por ejemplo, ¿es la función llevada a cabo?

La respuesta es a menudo la clave para entender en que atributo de calidad esta interesado el stakeholder que propuso el escenario. Si la respuesta a un estímulo de una función invocada por un usuario es simplemente que la función suceda, entonces el stakeholder probablemente esta interesado en la funcionalidad del sistema. Si el stakeholder le agrega "sin error" o "en dos segundos" al final, esto indica que esta interesado en reliability y performance, respectivamente. Notando cual es el atributo de calidad de interés, el líder de la evaluación le puede pedir al stakeholder que clarifique o refine el escenario si es necesario.

Tipos de escenarios:

En el ATAM hay tres tipos de escenarios: *escenarios de caso de uso* (típicamente usos del sistema existente), *escenarios de crecimiento* (cambios anticipados al sistema), y *escenarios exploratorios* (cambios extremos que se espera que estresen el sistema). Estos diferentes tipos de escenarios son utilizados para probar el sistema desde diferentes ángulos, optimizando las oportunidades de que aparezcan riesgos en las decisiones arquitectónicas.

Escenarios de Caso de Uso:

Los escenarios de caso de uso describen una interacción completa del usuario con el sistema corriendo. Por ejemplo, un usuario solicita un informe a una base de datos vía Web durante un período pico y lo recibe en cinco segundos (performance). Notar que los escenarios de caso de uso expresan deseos específicos de los stakeholders.

Escenarios de Crecimiento:

Los escenarios de crecimiento representan cambios típicos *anticipados* a un sistema. Cada escenario también tiene ramificaciones por atributos relacionados. Por ejemplo, el escenario "migrar a un sistema operativo de la misma familia, o a una nueva versión del sistema operativo existente con menos de un año-persona de trabajo", tiene implicaciones de performance, security y reliability.

Escenarios exploratorios:

Los escenarios exploratorios estresan el sistema. El objetivo de estos escenarios es exponer las condiciones límite del diseño actual, exponiendo posibles suposiciones implícitas. Los sistemas son raramente concebidos o diseñados para manejar esta clase de modificaciones, pero en el futuro estos podrían cambiar a requerimientos reales, entonces a los stakeholders les podría gustar entender las ramificaciones de dichos cambios. Por ejemplo, "cambiar la plataforma Unix subyacente a Macintosh".

Cada tipo de escenario (de caso de uso, de crecimiento y exploratorio) ayuda esclarecer los diferentes aspectos de la arquitectura, y juntos proveen una estrategia de evaluación de tres puntas. Los escenarios son un excelente vehículo con el cual todos los stakeholders comunican sus intereses arquitectónicos.

3.2.6 Paso 6: Analizar las propuestas arquitectónicas

En este momento, hay un conjunto priorizado de requerimientos de calidad concretos (del paso 5) y un conjunto de propuestas arquitectónicas utilizados en la arquitectura (del paso 4). Este paso mide cuan adecuados son el uno para el otro. Aquí, el equipo de evaluación puede probar las propuestas arquitectónicas que realizan los atributos

de calidad más importantes. Esto es hecho en detalle documentando estas propuestas arquitectónicas e identificando sus riesgos, no riesgos, sensitivity points y tradeoff points. La meta del equipo de evaluación es estar convencidos que la propuesta instanciada en la arquitectura que esta siendo evaluada, es la apropiada para satisfacer los requerimientos de un atributo específico.

Las salidas de este paso incluyen:

- Las propuestas arquitectónicas o decisiones relevantes para cada escenario de alta prioridad del árbol de utilidad.
- Las preguntas de análisis asociadas con cada propuesta, quedando los atributos de calidad equipados con los escenarios asociados.
- Las respuestas del arquitecto a las preguntas.
- Los riesgos, no riesgos, sensitivity points y tradeoff points identificados. Cada uno de estos esta asociado con el logro de uno o más refinamientos de los atributos de calidad en el árbol de utilidad.

En efecto, el árbol de utilidad le dice al equipo de evaluación donde probar la arquitectura, el arquitecto responde con la propuesta arquitectónica que satisface esta necesidad, y el equipo puede utilizar las preguntas sobre un atributo de calidad específico para probar la propuesta en mayor profundidad.

Las preguntas ayudan al equipo a:

- Entender la propuesta en detalle y como fue aplicada en esta instancia.
- Buscar las debilidades conocidas de la propuesta.
- Buscar los sensitivity points y los tradeoff points de la propuesta.
- Encontrar interacciones y concesiones mutuas (tradeoffs) con otras propuestas.

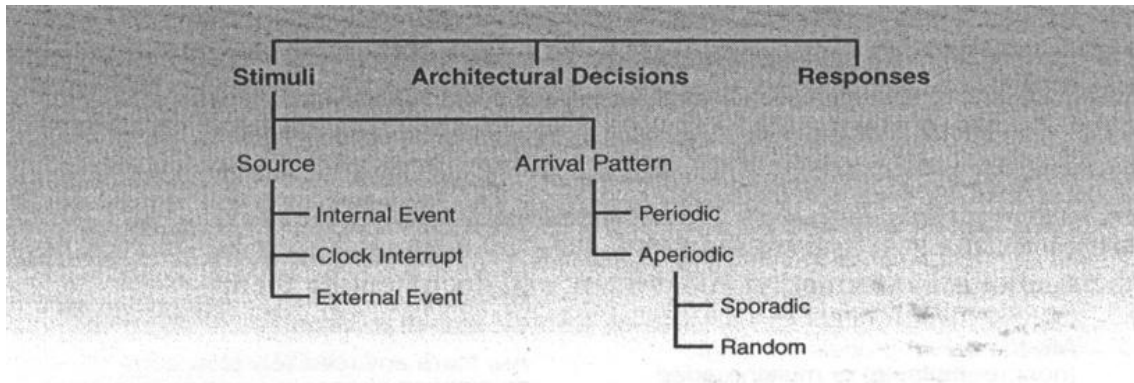
Al final, cada una de estas puede proveer el material básico para describir un riesgo, y registrarlo en una lista de riesgos que siempre crece.

Las preguntas de análisis no son un fin en sí mismas. Cada una es disparador de discusiones que determinarán un posible riesgo, no riesgo, sensitivity point o tradeoff point. El análisis no esta pensado para ser comprensivo y detallado. La clave esta en obtener la suficiente información arquitectónica para establecer un vínculo entre las decisiones arquitectónicas que han sido tomadas y los requerimientos de los atributos de calidad que necesitan ser satisfechos.

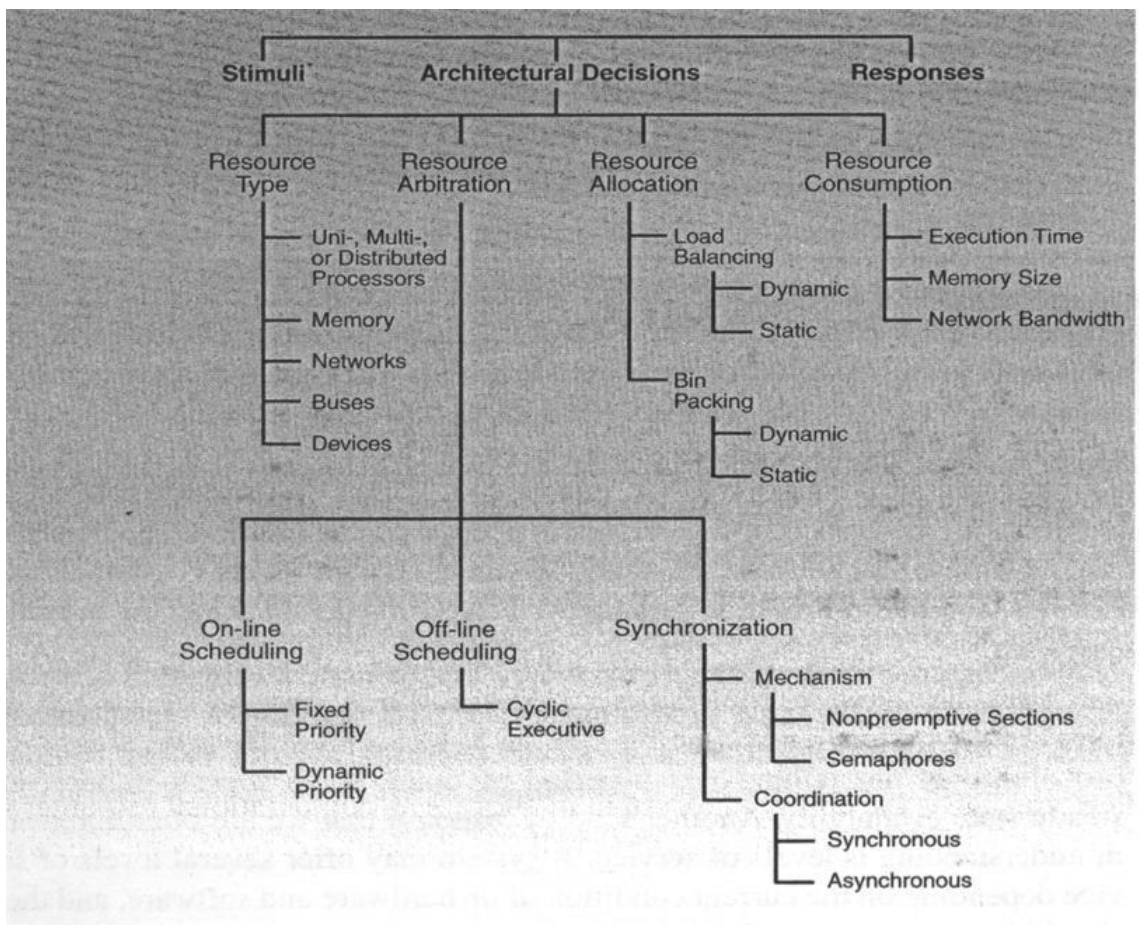
Todos los sensitivity points y tradeoff points son candidatos a riesgos. Al final del ATAM cada sensitivity point y tradeoff point debe ser catalogado como un riesgo o no riesgo. Los riesgos, no riesgos, sensitivity points y tradeoff points deben ser recolectados e indexados, en listas separadas.

Al final de este paso, el equipo de evaluación debería tener una idea clara de los aspectos más importantes de toda la arquitectura, la base lógica para las decisiones de diseño claves que han sido tomadas, y una lista de riesgos, no riesgos, sensitivity points y tradeoff points.

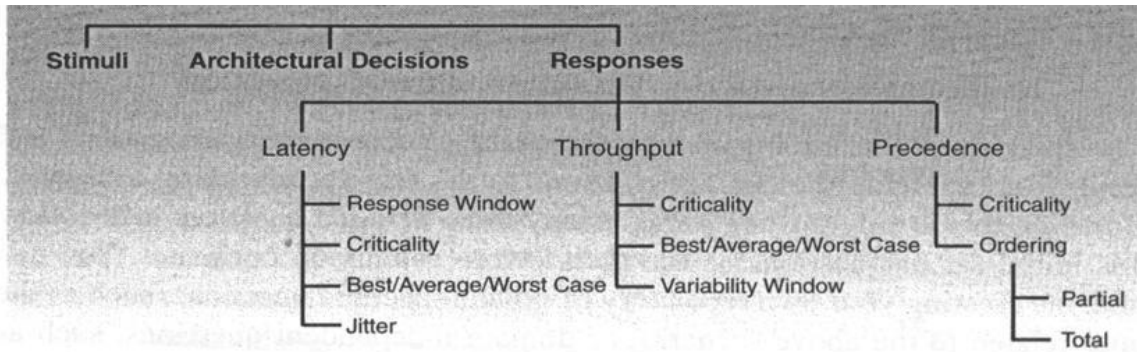
3.2.6.1 Ejemplos de caracterización de atributos de calidad



Caracterización de la Performance - Estímulo.



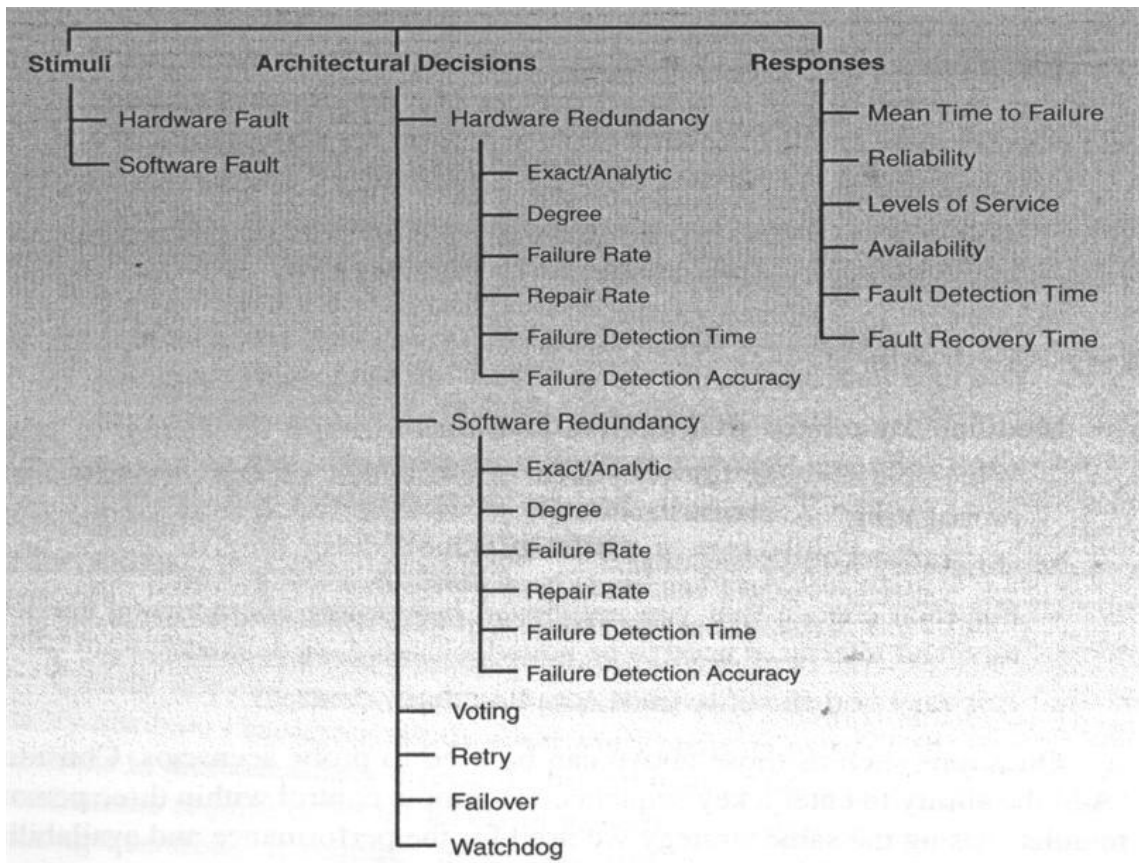
Caracterización de la Performance - Decisiones Arquitectónicas.



Caracterización de la Performance – Respuestas.

<p>Performance-Related Stimuli</p> <p>What is the client loading in terms of the number of concurrent sessions?</p> <p>How would you characterize the event model (periodic or stochastic)?</p> <p>What internal state changes trigger computations?</p> <p>Performance-Related Architectural Decisions</p> <p>Are the server processors single-(nonpreemptable) or multithreaded (preemptable)?</p> <p>How are (fixed or dynamic) priorities assigned to processes?</p> <p>Are processes statically or dynamically allocated to hardware?</p> <p>What is the physical distribution of the hardware and its connectivity?</p> <p>What are the bandwidth characteristics of the network?</p> <p>How is arbitration done in the network?</p> <p>Do you use synchronous or asynchronous protocols?</p>	<p>What is the location of firewalls and their impact on bandwidth?</p> <p>How are resources allocated to service requests?</p> <p>If there are multiple processes/threads competing for a shared resource, how are priorities assigned to these processes/threads and the process/thread controlling the shared resource?</p> <p>Are there any relatively slow communication channels along an important communication path (e.g., a modem)?</p> <p>Performance-Related Responses</p> <p>What is the performance impact of a thin versus a thick client in terms of latency or throughput?</p> <p>What are the performance characteristics of the middleware? What are the effects of load balancing and arbitration on throughput?</p> <p>Are there any computations that must be performed in a specific order?</p>
---	---

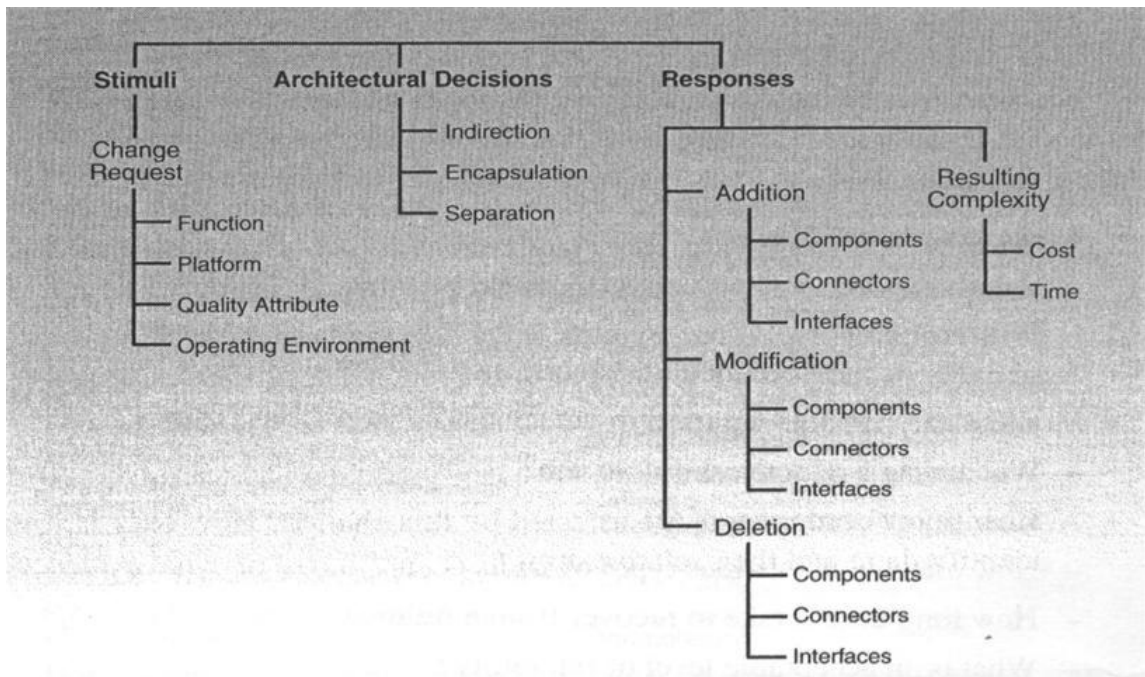
Ejemplos de preguntas de Performance.



Caracterización de la Availability.

<p>Availability-Related Stimuli</p> <p>How are hardware and software failures identified?</p> <p>Can active as well as passive failures be identified?</p> <p>If redundancy is used in the architecture, are there any common-mode failures?</p> <p>Availability-Related Architectural Decisions</p> <p>If redundancy is used in the architecture, what type of redundancy (analytic, replication, functional)?</p> <p>If redundancy is used in the architecture, how long does it take to switch between instances of a redundant component?</p> <p>If redundancy is used in the architecture, how many replicas of each critical component and connection are used?</p>	<p>How do you know that all critical components have been replicated?</p> <p>If live redundant components are used in the architecture, how does the system choose when their results differ?</p> <p>Does one component detect the failure of other components?</p> <p>Is there any kind of system "heartbeat" used to detect dead components?</p> <p>Availability-Related Responses</p> <p>Are there different levels of service available? How are these characterized?</p> <p>If a backing store is used, how quickly can a backup be made and how quickly can state be restored from the backup?</p>
---	---

Ejemplos de preguntas de Availability.



Caracterización de la Modifiability.

Modifiability-Related Stimuli	Modifiability-Related Responses
What new functionality do you anticipate adding in future versions of the system? Do you anticipate the need for making your system more secure in the future? How do you plan to handle migration to new releases of your operating system?	How many interfaces will change as a consequence of a change to some functionality? Do changes to your user interface affect application modules? What changes will result from adding a source of data such as a new sensor or user input field?
Modifiability-Related Architectural Decisions	What is the impact of a change in functionality on the ability to meet performance requirements?
Do any components have access to the implementation details of global variables? Do you use indirection mechanisms such as publisher/subscriber? How do you handle the possibility of changes in message formats?	How many components, connectors, and interfaces will be affected by removing obsolete data from the data repository?
Are implementation details or low-level services ever exposed to enhance performance?	How much will it cost to change from a uniprocessor to a multiprocessor platform? How long do you anticipate it will take to deploy your change?

Ejemplos de preguntas de Modifiability.

Analysis of Architectural Approach				
Scenario #: <i>Number</i>	Scenario: <i>Text of scenario from utility tree</i>			
Attribute(s)	<i>Quality attribute(s) with which this scenario is concerned</i>			
Environment	<i>Relevant assumptions about the environment in which the system resides, and the relevant conditions when the scenario is carried out</i>			
Stimulus	<i>A precise statement of the quality attribute stimulus (e.g., function invoked, failure, threat, modification . . .) embodied by the scenario</i>			
Response	<i>A precise statement of the quality attribute response (e.g., response time, measure of difficulty of modification)</i>			
Architectural Decisions	Sensitivity	Tradeoff	Risk	Nonrisk
<i>Architectural decisions relevant to this scenario that affect quality attribute response</i>	<i>Sensitivity Point #</i>	<i>Tradeoff Point #</i>	<i>Risk #</i>	<i>Nonrisk #</i>
...
...
Reasoning	<i>Qualitative and/or quantitative rationale for why the list of architectural decisions contribute to meeting each quality attribute requirement expressed by the scenario</i>			
Architectural Diagram	<i>Diagram or diagrams of architectural views annotated with architectural information to support the above reasoning, accompanied by explanatory text if desired</i>			

Plantilla para analizar una propuesta arquitectónica.

Analysis of Architectural Approach				
Scenario #: A12	Scenario: Detect and recover from HW failure of a primary CPU			
Attribute(s)	Availability			
Environment	Normal operations			
Stimulus	One of the CPUs fails			
Response	0.999999 availability of the switch			
Architectural Decisions	Sensitivity	Tradeoff	Risk	Nonrisk
Backup CPUs	S2		R8	
No backup data channel	S3	T3	R9	
Watchdog	S4			N12
Heartbeat	S5			N13
Failover routing	S6			N14
Reasoning	<ul style="list-style-type: none"> Ensures no common mode failure by using different hardware and operating system (see Risk R8) Worst-case rollover is accomplished in 4 seconds as computing state takes that long at worst Guaranteed to detect failure with 2 seconds based on rates of heartbeat and watchdog Watchdog is simple and proven reliable Availability requirement might be at risk due to lack of backup data channel (see Risk R9) 			
Architectural Diagram	<pre> graph LR In(()) --> P[Primary CPU OS1] In --> B[Backup CPU w/watchdog OS2] P -- heartbeat 1 sec. --> B P --> S[Switch CPU OS1] B --> S S --> Out(()) </pre>			

Ejemplo de un análisis de una propuesta arquitectónica.

3.2.7 Paso 7: Lluvia de ideas y priorización de escenarios

Los escenarios guían la fase de pruebas del ATAM. Está probado que generar un conjunto de escenarios facilita la discusión y la lluvia de ideas, cuando un gran número de stakeholders han sido reunidos para participar en el ATAM. Los escenarios son utilizados para:

- Representar los intereses de los stakeholders.
- Entender los requerimientos de un atributo de calidad.

Mientras que la generación del árbol de utilidad es primordial para entender como el arquitecto percibe y maneja las guías arquitectónicas de los atributos de calidad, el propósito de la lluvia de ideas de escenarios es tomarle el pulso a una gran comunidad de stakeholders. La lista priorizada de los escenarios de la lluvia de ideas es comparada con los generados por el árbol de utilidad. Si se descubren nuevos escenarios guías, estos también son una salida importante.

En este paso, el equipo de evaluación le pide a los stakeholders que piensen en los tres tipos de escenarios mencionados:

1. Los *escenarios de caso de uso* representan la manera en la cual los stakeholders esperan que el sistema sea utilizado. En los escenarios de caso de uso los stakeholders son usuarios finales.
2. Los *escenarios de crecimiento* representan la manera en la cual se espera que una arquitectura se adapte al crecimiento y al cambio en un tiempo cercano: modificaciones esperadas, cambios en la performance o availability, entre otros.
3. Los *escenarios de exploración* representan formas extremas de crecimiento, maneras en las cuales la arquitectura podría ser estresada por los cambios, por ejemplo, cambios dramáticos en la performance. Mientras que los escenarios de crecimiento muestran las fortalezas y debilidades de la arquitectura con respecto a fuerzas anticipadas en el sistema, los escenarios de exploración intentan encontrar más sensitivity points y tradeoff points que aparecen en los puntos de estrés de la arquitectura. La identificación de estos puntos ayuda a evaluar los límites de la arquitectura del sistema.

Una vez que los escenarios han sido recogidos, deben ser priorizados. Primero, se le pide a los stakeholders que unan los escenarios que consideren iguales. Luego, los stakeholders votan por los escenarios que crean más importantes.

En este punto de la evaluación, hacemos una pausa y comparamos el resultado de la priorización de los escenarios con el resultado del árbol de utilidad del paso 5, y se buscan acuerdos o desacuerdos. Cada escenario de alta prioridad de la lluvia de ideas es, en forma consensuada, ubicado en la hoja apropiada del árbol de utilidad.

Cuando un escenario de la lluvia de ideas es ubicado en el árbol de utilidad, una de estas tres cosas pasará:

1. El escenario coincidirá o prácticamente será un duplicado de uno ya existente en alguna hoja.
2. El escenario irá en una nueva hoja de una bifurcación existente.
3. El escenario no encajará en bifurcación alguna del árbol porque expresa un atributo de calidad que no fue considerado previamente.

El primer y segundo caso indican que la gran comunidad de stakeholders estuvo pensando en la misma línea que el arquitecto. El tercer caso, sin embargo, sugiere que el arquitecto pudo haber fallado en considerar un importante atributo de calidad, y subsecuentemente probar aquí podría producir fallas al documento.

La generación del árbol de utilidad y la lluvia de ideas de escenarios reflejan las metas de los atributos de calidad, pero por caminos de obtención diferentes y desde el punto de vista de los diferentes grupos de stakeholders.

Después de este paso, el árbol de utilidad continúa siendo el repositorio de todos los requerimientos de alta prioridad de los atributos de calidad

#	Scenario	Votes
4	<i>Dynamically replan a dispatched mission within 10 minutes</i>	28
27	<i>Split the management of a set of vehicles across multiple control sites</i>	26
10	<i>Change vendor analysis tools after mission has commenced without restarting system</i>	23
12	<i>Retarget a collection of diverse vehicles to handle an emergency situation in less than 10 seconds after commands are issued</i>	13
14	<i>Change the data distribution mechanism from CORBA to a new emerging standard with less than six person-months' effort</i>	12

Ejemplo de escenarios con ranking.

Scenario	# Votes	Quality Attributes
4	28	<i>Performance</i>
27	26	<i>Performance, modifiability, availability</i>
10	23	<i>Modifiability</i>
12	13	<i>Performance</i>
14	12	<i>Modifiability</i>

Escenarios con alto rango, con anotaciones de los atributos de calidad.

	Utility Trees	Scenario Brainstorming
Stakeholders	Architects, project leader	All stakeholders
Typical group size	evaluators; 2-3 project personnel	evaluators; 5-10 project-related personnel
Primary goals	Elicit, make concrete, and prioritize the driving quality attribute requirements Provide a focus for the remainder of the evaluation	Foster stakeholder communication to validate quality attribute goals elicited via the utility tree
Approach	General to specific: begin with quality attributes, refine until scenarios emerge	Specific to general: begin with scenarios, then identify quality attributes they express

Árbol de utilidad versus lluvia de ideas.

3.2.7.1 Stakeholders

Stakeholders for an Architecture Evaluation		
Stakeholder	Definition	Interest
Producers of the System		
Software architect	Person responsible for the architecture of the system and responsible for making tradeoffs among competing quality pressures	Moderation and mediation all of the quality concerns of the other stakeholders
Developer	Coder or designer	Clarity and completeness of architecture description, high cohesion of parts, limited coupling of parts, clear interconnection mechanisms
Maintainer	Person making changes after initial deployment	Maintainability, ability to locate every place in the system that must change in response to a modification
Integrator	Developer responsible for integrating (assembling) the components	Same as developer
Tester	Developer responsible for testing the system	Integrated, consistent error-handling protocols; limited component coupling; high component cohesion; conceptual integrity
Standards expert	Developer responsible for knowing the details of standards (current and future) to which the software must conform	Separation of concerns, modifiability, interoperability
Performance engineer	Person who analyzes system artifacts to see if the system will meet its performance and throughput requirements	Understandability, conceptual integrity, performance, reliability
Security expert	Person responsible for making sure that the system will meet its security requirements	Security
Project manager	Person who allocates resources to teams, is responsible for meeting schedule and budget, interfaces with customer	Clear structuring of architecture to drive team formation, work breakdown structure, milestones and deadlines, etc.

Stakeholders for an Architecture Evaluation (continued)		
Stakeholder	Definition	Interest
Product-line manager or "reuse czar"	Person who has a vision for how this architecture and related assets can be (re-) used to further the developing organization's long-range goals	Reusability, flexibility
Consumers of the System		
Customer	Purchaser of the system	Schedule of completion, overall budget, usefulness of the system, meeting customers' (or market's) expectations
End user	User of the implemented system	Functionality, usability
Application builder (in the case of a product-line architecture)	Person who will take the architecture and any reusable components that exist with it and instantiate them to build a product	Architectural clarity, completeness, simple interaction mechanisms, simple tailoring mechanisms
Mission specialist, mission planner	Representative of the customer who knows how the system is expected to be used to accomplish strategic objectives; has broader perspective than end users alone	Functionality, usability, flexibility
Servicers of the System		
System administrator	Person running the system (if different from user)	Ease in finding the location of problems that may arise
Network administrator	Person administering the network	Network performance, predictability
Service representatives	People who provide support for the use and maintenance of the system in the field	Usability, serviceability, tailorability
Persons Interfacing or Interoperating with the Software		
Representatives of the domain or community	Builders or owners of similar systems or systems with which the subject system is intended to work	Interoperability

Stakeholders for an Architecture Evaluation (<i>continued</i>)		
Stakeholder	Definition	Interest
System architect	Architect of the entire system; person who makes tradeoff decisions between hardware and software and who selects the hardware environment	Portability, flexibility, performance, efficiency
Device expert	Person who knows the devices with which the software must interface; can predict future trends in hardware technology	Maintainability, performance

Stakeholders de la evaluación de la arquitectura.

3.2.8 Paso 8: Analizar las propuestas arquitectónicas

Después que los escenarios han sido recolectados y analizados, el equipo de evaluación guía al arquitecto en el proceso de llevar a cabo los escenarios con ranking más alto del paso 7 en todas las descripciones arquitectónicas que se han presentado. El arquitecto explica como las decisiones arquitectónicas relevantes contribuyen a realizar el escenario.

En este paso el equipo de evaluación realiza las mismas actividades que en el paso 6, mapeando los escenarios recientemente generados con ranking más alto en los artefactos arquitectónicos. Como este es un paso de prueba, se espera que poca información sea descubierta.

3.2.9 Paso 9: Presentar los resultados

Finalmente, la información recogida por el ATAM necesita ser resumida y presentada a los stakeholders. En esta presentación el líder de la evaluación recapitula los pasos del ATAM y toda la información recogida en cada paso, incluyendo el contexto del negocio, los requerimientos guías, las restricciones, y la arquitectura. Pero más importante son las salidas del ATAM:

- El documento de propuestas arquitectónicas.
- El conjunto de escenarios priorizados.
- El conjunto de preguntas basadas en los atributos.
- El árbol de utilidad.
- Los riesgos descubiertos.
- Los no riesgos documentados.
- Los sensitivity points y tradeoff points encontrados.

Todas estas salidas son descubiertas, capturadas y catalogadas durante la evaluación. Pero en este paso, el equipo de evaluación produce una salida adicional: temas de riesgo. Los riesgos pueden ser agrupados basados en algunas preocupaciones subyacentes comunes o deficiencias sistemáticas. Por cada tema de riesgo, el equipo de evaluación identifica cuales pautas del negocio, listadas en paso 2, son afectadas. Identificar temas de riesgo y relacionarlos con pautas específicas produce dos efectos. Primero, cierra el círculo de la evaluación relacionando los resultados finales con la presentación inicial. Segundo, eleva los riesgos que se descubrieron a la atención de la dirección del proyecto.

Como el equipo de evaluación esta sistemáticamente trabajando en intentar de entender las propuestas arquitectónicas, es inevitable que, a veces, haga recomendaciones de como la arquitectura podría haber sido diseñada, o analizada diferente. Estas estrategias de mitigación podrían estar relacionadas con el proceso, podrían ser directivas, o podrían ser técnicas. Sin embargo, no es una parte integral del ATAM ofrecer estrategias de mitigación. El objetivo del ATAM es localizar los riesgos de la arquitectura.

3.2.9.1 Sensitivities, riesgos y no riesgos

El ATAM confía fuertemente en la identificación de los sensitivity points y los riesgos de la arquitectura. Confía en que los sensitivity points no solo descubrirán potenciales problemas (riesgos), sino también fortalezas de la arquitectura. Un aspecto del método es que no solo encuentra riesgos, sino que también encuentra no riesgos.

Recordar que los no riesgos, como los riesgos, están relacionados con las respuestas arquitectónicas provenientes de las decisiones arquitectónicas. Pero con los no riesgos se puede decir que las decisiones arquitectónicas son apropiadas, es decir, el diseño de la arquitectura satisface los requerimientos de los atributos calidad. Interesa registrar esta información, como se registra información sobre los riesgos, porque si esta decisión arquitectónica alguna vez cambia, es necesario examinar si el no riesgo se transforma en un riesgo.

Como es sabido, un sensitivity point es una propiedad de uno o más componentes (y/o relaciones entre componentes) que son críticos para lograr una respuesta de un atributo de calidad particular. ATAM, sugiere que cada sensitivity point sea clasificado como riesgo o no riesgo, dependiendo si la respuesta deseada es lograda o no.

3.3 Las fases del ATAM

En esta sección describiremos como los pasos del ATAM se distribuyen en el tiempo. El ATAM esta compuesto por cuatro fases, que corresponden a los intervalos de tiempo de mayor actividad.

La fase 0 es de preparación, en la cual el equipo de evaluación es creado y se forma una sociedad entre la organización evaluadora y la organización cuya arquitectura será evaluada. Las fases 1 y 2, las fases de evaluación del ATAM, comprenden los nueve pasos vistos hasta ahora. La fase 1 esta centrada en la arquitectura, y se concentra en obtener y analizar la información arquitectónica. La fase 2 esta centrada en los stakeholders, y se concentra en obtener los puntos de vista de los de los stakeholders y verificar los resultados de la fase 1. En la fase 3 se realiza el reporte final, y se continua con las acciones (si hay) planeadas, y la organización evaluadora actualiza sus archivos e incorpora la experiencia adquirida.

Outputs of the ATAM:

Steps of ATAM:	Prioritized Statement of Quality Attribute Requirements.	Catalog of Architectural Approaches Used	Approach- and Quality-Attribute-Specific Analysis Questions	Mapping of Architectural Approaches to Quality Attributes	Risks and Nonrisks	Sensitivity and Tradeoff Points
1. Present the ATAM						
2. Present business drivers	*a				*b	
3. Present architecture		**			*c	*d
4. Identify architectural approaches		**	**		*e	*f
5. Generate quality attribute utility tree	**					
6. Analyze architectural approaches		*g	**	**	**	**
7. Brainstorm and prioritize scenarios	**					
8. Analyze architectural approaches		*g	**	**	**	**
9. Present results						

Key: ** = the step is a primary contributor to the output; * = the step is a secondary contributor.

- The business drivers include the first, coarse description of the quality attributes.
- The business drivers presentation might disclose an already-identified or long-standing risk which should be captured.
- The architect may identify a risk in his or her presentation.
- The architect may identify a sensitivity or tradeoff point in his or her presentation.
- Many architectural approaches have standard risks associated with them.
- Many architectural approaches have standard sensitivities and quality attribute tradeoffs associated with them.
- The analysis steps might reveal one or more architectural approaches not identified in Step 4, which will then produce new approach-specific questions.

Tabla de pasos y salidas del ATAM.

3.3.1 Fase 0

En las fases 1 y 2 del ATAM es donde se realiza el análisis y por eso son consideradas el "corazón" del método. Pero antes que la fase 1 pueda comenzar, se debe establecer una sociedad entre el responsable de la evaluación y la organización que la lleva adelante. Los propósitos de la fase 0 son establecer acuerdos sobre tiempo, fechas, costos, y disposición al trabajo, y formar el equipo de evaluación.

En la fase 0 se realiza todo el trabajo base para que la evaluación arquitectónica pueda comenzar, y asegurar su éxito.

La fase 0 consiste de dos partes: establecer una sociedad con el cliente y la preparación para las siguientes fases.

3.3.1.1 Sociedad

La fase 0 involucra comunicarse con las personas que solicitan la evaluación, los clientes.

El cliente debe ser alguien que pueda ejercer control sobre el proyecto cuya arquitectura es objeto de evaluación, por ejemplo, el cliente podría ser el director del proyecto. Se asume que el cliente tiene suficiente influencia para lograr que el desarrollo se tome una pausa, para que la arquitectura pueda ser evaluada. Y también, se asume que el cliente tiene acceso a una amplia selección de stakeholders para la arquitectura.

Los siguientes problemas deben ser resueltos antes de que el cliente de el visto bueno para que la evaluación comience:

1. El cliente debería tener un conocimiento básico del método y del proceso que se sigue. Esto se puede realizar mediante una charla informativa o una descripción escrita del método.
2. El cliente debería describir el sistema y la arquitectura que es evaluada. Esto permite al líder de la evaluación decidir si hay, o no, suficiente material para que una evaluación basada en ATAM sea útil. En este punto, el líder de la evaluación debe decidir si se continua, o no, con la evaluación.
3. En caso de continuar, se debería negociar y firmar un contrato de trabajo. Esto asegura a ambos involucrados que los siguientes puntos están entendidos:
 - Quien es responsable de proveer los recursos necesarios para la evaluación.
 - A quien corresponde el informe final de la evaluación y cuando hay que entregárselo.
 - Cual es la disponibilidad del equipo para continuar el trabajo.
4. Los problemas de propiedad de la información deberían ser resueltos.

La fase de negociación también es un buen momento para conversar sobre el costo/beneficio de la evaluación arquitectónica.

3.3.1.2 Preparación

La preparación de la fase 0 consisten en:

- Formar el equipo de evaluación.
- Tener una reunión inicial con el equipo de evaluación.
- Hacer las preparaciones necesarias para la fase 1.

Una vez que el equipo esta armado, es necesario asignarle a cada miembro un rol en la evaluación venidera. Es una buena idea rotar los roles entre los miembros del equipo, como forma de capacitar a todos los miembros del equipo en todos los roles.

Un rol puede ser asumido por más de una persona, y una persona puede asumir más de un rol. Es el líder de la evaluación quien asigna las personas a los roles, y los roles a las personas. Algunas reglas que ayudan:

- El mínimo de integrantes de un equipo de evaluación debería ser cuatro personas.
- Una persona puede desempeñar los roles de observador del proceso, controlador del tiempo e interrogador.
- El líder del equipo generalmente también es el líder de la evaluación.
- El interrogador debería ser especializado en las cualidades que interesan del sistema.

Role	Responsibilities	Desirable Characteristics
Team leader	Sets up the evaluation; coordinates with client; makes sure client's needs are met; establishes evaluation contract. Forms the evaluation team. In charge of seeing that final report is produced and delivered (although the writing may be delegated).	Well-organized, with managerial skills. Good at interacting with client. Able to meet deadlines.
Evaluation leader	Runs evaluation. Facilitates elicitation of scenarios; administers scenario selection/prioritization process; facilitates evaluation of scenarios against architecture. Facilitates on-site analysis.	Comfortable in front of an audience. Excellent facilitation skills. Good understanding of architectural issues. Practiced in architecture evaluations. Able to tell when protracted discussion is leading to a valuable discovery, or when it is pointless and should be redirected.
Scenario scribe	Writes scenarios on flip chart or whiteboard during scenario elicitation process. Carefully captures agreed-upon wording of each scenario and doesn't let discussion continue until exact wording is captured.	Good handwriting. Willingness to be a stickler about not moving on before an idea (a scenario) is captured. Able to quickly absorb and distill the essence of technical discussions.
Proceedings scribe	Captures the proceedings in electronic form on a laptop computer or in-room workstation. Captures the raw scenarios. Captures the issue(s) that motivated each scenario, as this is often lost in the wording of the scenario itself. Captures the resolution of each scenario when applied to architecture(s). Generates a printed list of adopted scenarios for hand-out to all participants.	Good, fast typist. Well-organized to allow rapid recall of information. Good understanding of architectural issues. Able to assimilate technical issues quickly. Must not be afraid to interrupt the flow of discussion (at opportune times) to test understanding of an issue, so that the appropriate information is captured.

Role	Responsibilities	Desirable Characteristics
Timekeeper	Helps the evaluation leader stay on schedule. Helps control amount of time devoted to each scenario during evaluation phase.	Willingness to brazenly interrupt discussion to call time.
Process observer	Keeps notes on where the evaluation process itself could be improved or deviated from the plan. Usually a silent observer, but may make process-based suggestions to the evaluation leader, discretely, during the evaluation. After evaluation, reports on how the process went and what lessons were learned for future improvement. Also responsible for reporting experience to architecture evaluation team at large.	Thoughtful observer. Knowledgeable in the evaluation process. Should have previous experience in the architecture evaluation method.
Process enforcer	Helps the evaluation leader remember and carry out the steps of the evaluation method.	Should be fluent in the steps of the method. Willing and able to provide guidance to the evaluation leader in a discrete manner.
Questioner	Raises issues of architectural interest that perhaps the stakeholders have not considered.	Good architectural insights; good insights into needs of stakeholders. Experience with systems in similar domain. Not afraid to bring up possibly contentious issues and pursue them doggedly. Familiarity with attributes of concern.

Tabla de roles y responsabilidades del equipo de evaluación.

Después que el equipo esta formado, se debería hacer una reunión inicial donde se intercambie todo el conocimiento sobre la evaluación y los roles sean asignados.

Preparaciones para la fase 1 incluyen tomar recaudo de una infinidad de detalles logísticos para asegurar que todos estén en el momento y lugar correcto preparados para trabajar, esto incluye, además del equipo de evaluación, a los representantes del proyecto.

3.3.2 Fase 1

En la fase 1, el equipo de ATAM se encuentra con una parte del equipo cuya arquitectura esta siendo evaluada, de repente por primera vez. Esta reunión tiene dos objetivos: organizar el resto de la evaluación y recolectar información.

Con un pequeño grupo de de personas claves, la fase 1 se concentra en los pasos 1 al 6. El equipo de evaluación presenta el ATAM, un portavoz del proyecto presenta las pautas del negocio, y el arquitecto presenta la arquitectura. El grupo cataloga las propuestas arquitectónicas y construye el árbol de utilidad. Los escenarios de alta prioridad del árbol de utilidad serán luego la base del análisis.

Además de realizar los seis pasos, el equipo de evaluación tiene otros objetivos para cumplir durante la fase 1. Necesita obtener tanta información como sea posible para determinar:

- Si el resto de la evaluación es factible y debería continuar. Si no, la fase 1 es un buen lugar para cortar.
- Si se necesita más documentación de la arquitectura, y de ser así, determinar que tipo de documentación y como debería ser presentada.
- Cuales stakeholders deberían estar presentes en la fase 2. Al final de la fase 1 el responsable de la evaluación se debe asegurar que los stakeholders adecuados sean convocados a la fase 2.

Hay un intervalo entre las fases 1 y 2 en donde el equipo de arquitectos continúa con el análisis, en colaboración con el equipo de evaluación. En esta fase el equipo de evaluación no construye modelos analíticos detallados, pero construye modelos rudimentarios que le darán ideas a los evaluadores y al arquitecto sobre la arquitectura, para hacer que la reunión de la fase 2 sea productiva. Además, en este intervalo el equipo de evaluación toma su forma final, de acuerdo a las necesidades de la evaluación. Por ejemplo, si el sistema es centrado en la base de datos, un experto en diseño de base de datos podría ser reclutado para el equipo.

3.3.3 Fase 2

A esta altura, el equipo de evaluación habrá entendido con suficiente detalle la arquitectura como para verificar el análisis ya realizado, y llevar a cabo el análisis necesario restante. Los stakeholders apropiados fueron identificados, y se les dio material de lectura avanzado, como por ejemplo una descripción del ATAM y documentación del sistema. Ahora, los stakeholders son reunidos para la fase 2, que puede involucrar tres a cuarenta stakeholders (lo recomendado es no más de 15).

Como habrá un mayor número de stakeholders asistiendo a la fase 2 y como las reuniones entre la fase 1 y la fase 2 están separadas por días o semanas, la fase 2 comienza con una repetición del paso 1. Luego, se hace una recapitulación de los pasos 2 al 6 de la fase 1. La evaluación continúa ejecutando los pasos 7, 8 y 9.

Step	Activity	Participants for Phase 1	Participants for Phase 2
1	Present the ATAM	Evaluation team and project decision makers	Evaluation team, project decision makers, and all stakeholders
2	Present business drivers		
3	Present architecture		
4	Identify architectural approaches		
5	Generate quality attribute utility tree		
6	Analyze architectural approaches		
7	Brainstorm and prioritize scenarios	N/A	
8	Analyze architectural approaches		
9	Present results		

Tabla con los pasos del ATAM asociados a los grupos de stakeholders.

A continuación se presenta una posible agenda para las fases 1 y 2. Es un ejemplo de como planificar las actividades a realizar, no pretende ser seguido en forma estricta.

Start	Activity	
8:30 am	Introductions/ATAM Presentation (Step 1)	Phase 1
10:00	Present Business Drivers (Step 2)	
10:45	Break	
11:00	Present Architecture (Step 3)	
12:00	Identify Architectural Approaches (Step 4)	
12:30 pm	Lunch	
1:45	Generate Quality Attribute Utility Tree (Step 5)	
2:45	Analyze Architectural Approaches (Step 6)	
3:45	Break	
4:00	Analyze Architectural Approaches (Step 6)	
5:00 pm	Adjourn	
		Hiatus
Day 1		Phase 2
8:30 am	Introductions/ATAM Presentation (Step 1)	
9:15	Present Business Drivers (Step 2)	
10:00	Break	
10:15	Present Architecture (Step 3)	
11:15	Identify Architectural Approaches (Step 4)	
12:00	Lunch	
1:00 pm	Generate Quality Attribute Utility Tree (Step 5)	
2:00	Analyze Architectural Approaches (Step 6)	
3:30	Break	
3:45	Analyze Architectural Approaches (Step 6)	
5:00 pm	Adjourn for the Day	
Day 2		
8:30 am	Introductions/Recap ATAM	
8:45	Analyze Architectural Approaches (Step 6)	
9:30	Brainstorm Scenarios (Step 7)	
10:30	Break	
10:45	Prioritize Scenarios (Step 7)	
11:15	Analyze Architectural Approaches (Step 8)	
12:30 pm	Lunch	
1:30	Analyze Architectural Approaches (Step 8)	
2:45	Prepare Presentation of Results/Break	
3:30	Present Results (Step 9)	
5:00 pm	Adjourn	

Ejemplo de agenda para las fases 1 y 2 del ATAM.

3.3.4 Fase 3

Al término del ATAM, el informe final debe ser escrito y entregado. Pero también es importante mantener la capacidad del ATAM actualizando los repositorios de artefactos, encuestas y medidas de esfuerzo tomadas, y los cuestionarios realizados al equipo de evaluación para encontrar formas de mejorar el método.

En la fase 3, las siguientes tareas deben ser cumplidas:

1. Realizar el informe final.
2. Recoger información para la medida y mejora del proceso.
3. Actualizar los repositorios de artefactos.

3.3.4.1 Realizar el informe final

El informe final se realiza en la fase 3. Producir el informe final significa catalogar: a) ¿qué se hizo?, b) ¿qué se encontró?, c) ¿qué se concluyó?. Al asignar la responsabilidad a miembros del equipo de secciones específicas del informe, al comienzo de la evaluación, tiene como resultado que escribirlo se hace rápido y eficiente.

3.3.4.2 Recoger información para la medida y mejora del proceso

Cada evaluación provee una conveniente oportunidad para recoger información de manera tal de mejorar las ideas sobre costo/beneficio de efectuar evaluaciones, y también recoger opiniones sobre lo que anduvo bien y lo que anduvo mal.

La información tiene dos fuentes: el equipo de evaluación y el cliente. En ambos casos de debe recoger información sobre mejora y costo/beneficio. En particular, se recomienda realizar las encuestas a los clientes luego de seis meses de realizada la evaluación, con el objetivo de medir los efectos a largo plazo.

Se recomienda realizar las siguientes cinco encuestas:

1. Una encuesta de mejora a los participantes, preguntándoles sus impresiones sobre la evaluación realizada.
2. Una encuesta de mejora a los miembros del equipo, preguntándoles sus impresiones sobre la evaluación realizada.
3. Una encuesta de costo al cliente.
4. Una encuesta de costo al equipo de evaluación.
5. Una encuesta de beneficio a largo plazo al cliente.

3.3.4.3 Actualizar los repositorios de artefactos

Se recomienda mantener repositorios de los artefactos utilizados y producidos durante cada evaluación. Esto ayudará en futuras evaluaciones.

Además de registrar información de costo/beneficio, es conveniente guardar también los escenarios producidos. En el futuro sistemas similares pueden ser evaluados, entonces estos escenarios pueden servir de mucha ayuda a la hora de generar los nuevos escenarios. Es más, a medida que los distintos escenarios se van guardando, se puede generar una checklist, para luego utilizarla en cada arquitectura a evaluar.

Aparte de los escenarios, es conveniente realizar una lista de las preguntas de análisis utilizadas, estas son las mejores herramientas del equipo de evaluación.

También es importante agregar al repositorio los comentarios de los participantes. Futuros líderes de evaluaciones pueden leerlos y obtener valiosas ideas de pasadas evaluaciones.

Finalmente, archivar una copia del informe final. Futuros equipos de evaluación apreciarán tener una plantilla para hacer sus informes.

4 CONCLUSIONES

4.1 ¿Qué hemos visto de ATAM?

Los escenarios fueron el punto de partida del desarrollo de ATAM. El ATAM nace de la noción de las concesiones mutuas entre los diferentes atributos de calidad. El árbol de utilidad es una buena herramienta para generar escenarios. Realizando una recorrida top – down del mismo, se asegura cubrir todos los atributos de calidad claves. El ATAM también saca ventaja del hecho de que para poder lograr las metas de los atributos de la calidad, debe existir sinergia entre las decisiones arquitectónicas. El análisis se realiza mapeando los escenarios en la arquitectura, en base a la obtención y estudio de las propuestas arquitectónicas. ATAM también relaciona en forma explícita a los riesgos arquitectónicos y a los tradeoffs (concesiones mutuas) con las pautas del negocio.

4.2 ¿Por qué evaluar arquitecturas?

Si las arquitecturas de software son importantes, entonces también lo son las evaluaciones de las arquitecturas de software.

La mayor meta de la evaluación es obtener los riesgos arquitectónicos. En la mayoría de los casos, los riesgos nunca se escriben de una manera concreta. Antes de la evaluación los riesgos son una colección abstracta, difusa y amorfa. Luego de la evaluación son concretos, documentados y manejables.

Una de las particularidades centrales de ATAM es que comienza con las pautas del negocio, y termina con las pautas del negocio. Las pautas del negocio motivan las metas de los atributos de calidad, en las cuales se centra la evaluación de la arquitectura. La evaluación produce riesgos, los cuales están ligados a las pautas del negocio sobre las cuales potencialmente tengan un impacto. El poder ser capaz de concluir que una colección particular de riesgos arquitectónicos podría poner en peligro el éxito de una importante meta del negocio es simplemente muy poderoso, porque permite simultáneamente que el director tome medidas preventivas y los ingenieros actúen.

4.3 ¿Por qué ATAM funciona?

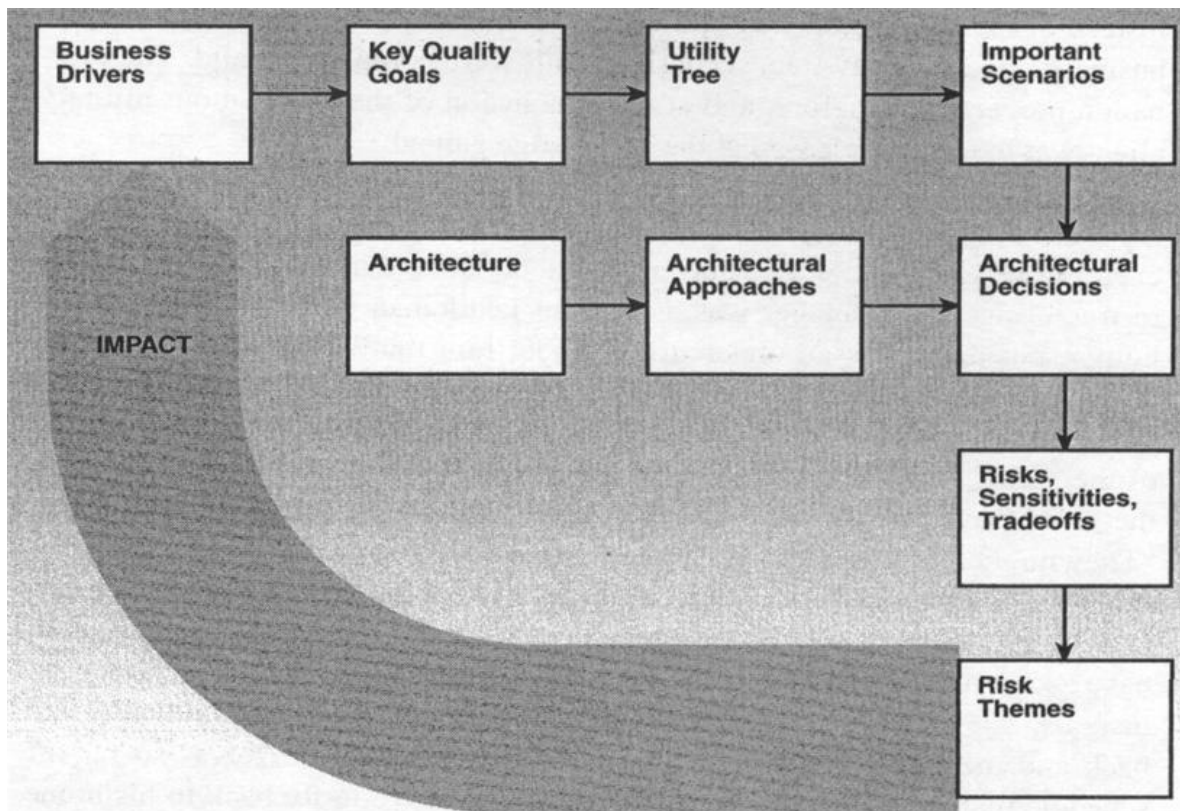
El ATAM fue diseñado para obtener los requerimientos de los atributos de calidad que son importantes para lograr las metas del negocio. La expresión concreta de esas metas son los escenarios de los atributos de calidad que aparecen en las hojas del árbol de utilidad.

El ATAM también fue diseñado para obtener las propuestas arquitectónicas utilizadas para alcanzar las metas de los atributos de calidad. Los escenarios y las propuestas se unen cuando se pregunta como los escenarios más importantes son soportados por la arquitectura. El resultado es un conjunto de decisiones arquitectónicas. Si estas decisiones son potencialmente problemáticas o especialmente importantes, o si afectan a más de un atributo, entonces son registradas como riesgos, sensitivity points y/o tradeoff points. Los riesgos son agrupados en temas de riesgo, los cuales son examinados de acuerdo a su impacto en lograr las pautas del negocio. Por lo tanto, el ATAM expone los riesgos arquitectónicos que potencialmente prohíben a una organización conquistar las metas del negocio.

Las propiedades del ATAM que realzan su habilidad para encontrar riesgos arquitectónicos significativos son las siguientes:

1. *Alcance auto contenido de las pautas del negocio.* El ATAM se centra en la evaluación de aquellos aspectos de la arquitectura que tengan un impacto significativo en lograr las metas del negocio.
2. *Auto escalado basado en el número de escenarios analizados.* El ATAM puede llevarse a cabo tanto en un período corto como extenso de tiempo. LA diferencia estará en el número de escenarios que serán analizados.
3. *Involucramiento de todos los stakeholders.* El ATAM esta diseñado para obtener información de todos los stakeholders relevantes, y explotar sus talentos y perspectivas.
4. *El ATAM insta al equipo de evaluación a mejorar el proceso en base a la experiencia.*

En resumen, el ATAM apunta a los requerimientos de los atributos de calidad esenciales utilizando las pautas del negocio como clave para el criterio de priorización. Al mismo tiempo, se concentra en las decisiones arquitectónicas principales explotando la experiencia del equipo de evaluación y el arquitecto. Finalmente, el ATAM ofrece un entendimiento de las decisiones arquitectónicas más importantes de las ramificaciones del negocio.



Flujo conceptual del ATAM.

5 REFERENCIAS

- Evaluating Software Architectures: Methods and Case Studies; Paul Clements, Rick Kazman y Mark Klein; Marzo 2004; ISBN 0-201-70482-X.
- Steps in an Architecture Tradeoff Analysis Method: Quality Attribute Models and Analysis; Mario R. Barbacci, S. Jeromy Carriere, Peter H. Feiler, Rick Kazman, Mark H. Klein, Howard F. Lipson, Thomas A. Longstaff y Charles B. Weinstock; Mayo 1998.
- Architecture-Based Development; Len Bass y Rick Kazman; Abril 1999.
- ATAM: Method for Architecture Evaluation; Rick Kazman, Mark Klein y Paul Clements; Agosto 2000.
- Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study; Lawrence G. Jones y Anthony J. Lattanze; Diciembre 2001.
- Evaluación de Arquitecturas de Software; Dieter Spangenberg, Gustavo Signorele, Maximiliano Panario, Flavio Spagnuolo y Sebastián Lacuesta; Junio 2003.
- Evaluación Arquitectónica de un Software Basado en Componentes; Maria Pérez, Luís Mendoza, Dakar Parada y George Di Paula; 2004.