

PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 07-01

**Evaluación de Arquitecturas de Software
con ATAM (Architecture Tradeoff Analysis
Method): un caso de estudio**

Andrea Delgado Alberto Castro Martín Germán

2007

Metodología para desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture)

Delgado, Andrea; Castro, Alberto; Germán, Martín

ISSN 0797-6410

Reporte Técnico **RT 07-01**

PEDECIBA

Instituto de Computación – Facultad de Ingeniería

Universidad de la República

Montevideo, Uruguay, 2007

1 Evaluación de Arquitecturas de Software con ATAM (Architecture Tradeoff Analysis Method): un caso de estudio [‡]

Andrea Delgado, Alberto Castro, Martín Germán
*Universidad de la República,
Instituto de Computación,
Grupo de Ingeniería de Software,
Montevideo, Uruguay
{adelgado, acastro, mgerman}@fing.edu.uy*

Febrero 2007

Resumen

La Arquitectura de Software condiciona las características del producto final en cuanto a cualidades como desempeño y mantenibilidad. El Architecture Tradeoff Analysis Method (ATAM) es una metodología para evaluar Arquitecturas de Software basada en los atributos de calidad especificados para el sistema, desarrollada por el Software Engineering Institute (SEI). El Grupo de Ingeniería de Software (Gris) del Instituto de Computación tiene como eje de sus actividades un programa de construcción y prueba de modelos de proceso, en cuyo contexto se desarrolló una herramienta para especificar modelos de procesos. ATAM se puso en práctica para evaluar el producto obtenido y realizar un segundo ciclo de desarrollo sobre el mismo, mejorando aspectos de riesgo identificados principalmente para el atributo de calidad performance. Se presenta ATAM y el caso de estudio realizado en el marco del programa.

Palabras clave: Ingeniería de Software, Arquitecturas de Software, Metodologías de evaluación de Arquitecturas de Software, Architecture Tradeoff Analysis Method (ATAM), atributos de calidad

[‡] Artículo publicado en las “VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC’07)”, Lima, Perú, Febrero de 2007. ISBN 978-9972-2885-1-7

1 Introducción

El eje de las actividades del grupo de Ingeniería de Software (Gris) del Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República, Uruguay [1] está constituido por el programa de construcción y prueba de modelos de proceso. El programa inició en el año 2000 y utiliza como banco de pruebas de los procesos al curso “Proyecto de Ingeniería de Software” [2] que se dicta en cuarto año de la carrera de Ingeniería en Computación. En este curso los estudiantes desarrollan aplicaciones en proyectos sometidos a restricciones similares a las de la industria, contrastando la teoría con la práctica, donde clientes externos al curso plantean requerimientos para la obtención de aplicaciones de mediano porte con desafíos tecnológicos importantes. Si bien el objetivo central del programa es el desarrollo de modelos de proceso que puedan resultar adecuados para su transferencia a la industria, también se utiliza para desarrollar prototipos de aplicaciones en el marco de proyectos del Instituto de Computación y de la Facultad de Ingeniería, así como para empresas del medio. Una descripción completa del programa desde su concepción y evolución hasta el presente puede ser consultada en [3].

En el contexto del programa, en el año 2004 un docente del grupo de Ingeniería de Software (Gris) presentó un proyecto para la construcción de una herramienta que permitiera especificar modelos de procesos y generar en base a dicha especificación un documento con la descripción del proceso y un sitio web con la definición realizada, utilizando para las pruebas el proceso base del programa que es una adaptación del Rational Unified Process (RUP) [4]. Para el año 2005 se decidió realizar un segundo ciclo de desarrollo para evolucionar el prototipo de la herramienta obtenido, mejorando principalmente aspectos de performance, y agregando funcionalidades nuevas. En marzo del 2005 se propone por parte de otro docente del grupo de Ingeniería de Software (Gris), una pasantía para realizar la evaluación de la herramienta obtenida aplicando el Architecture Tradeoff Analysis Method (ATAM) [5] desarrollado por el Software Engineering Institute (SEI) [6], de forma de identificar los aspectos de riesgo que se deberían tener en cuenta en este segundo ciclo de desarrollo, y maximizar la reutilización de productos generados por dicho proyecto, además de evaluar la metodología de evaluación propuesta en ATAM como parte del estudio de la pasantía [7].

2 Architecture Tradeoff Analysis Method (ATAM)

El Architecture Tradeoff Analysis Method (ATAM) es una metodología para evaluar Arquitecturas de Software que principalmente evalúa la adecuación de la Arquitectura de Software definida con respecto a los atributos de calidad especificados para el sistema. Surge confluendo ideas y técnicas de tres áreas: la noción de estilos o patrones de arquitectura, el análisis de atributos de calidad y el método Software Architecture Analysis Method (SAAM) [8] que es el predecesor del ATAM. Obtiene su nombre del concepto de que a veces no será posible cumplir con todos los atributos de calidad definidos y se deberán realizar concesiones mutuas (o tradeoffs) entre estos, para obtener el balance adecuado.

Existen también otros métodos para analizar arquitecturas como el Quality Attribute Workshop (QAW) [9], que surge para complementar el ATAM, y que identifica atributos de calidad importantes y clarifica los requerimientos del sistema antes de que exista una arquitectura definida que pueda ser evaluada con ATAM, y que si bien resulta más simple de aplicar, no es tan completo. Los métodos de evaluación de arquitecturas son bastante recientes, la Arquitectura de Software como tal surge en la década de los 90 y se populariza como área de investigación y práctica dentro de la Ingeniería de Software debido principalmente al crecimiento en la complejidad de los sistemas de software actuales incluyendo la variedad de nuevas tecnologías disponibles. Esto hace que cada vez sea más necesaria la definición explícita y documentación de la Arquitectura de Software, teniendo en cuenta la identificación de estilos o patrones de arquitecturas [10] que son utilizados comúnmente y presentan características específicas, y cuya aplicación trae aparejadas determinadas ventajas y desventajas en el contexto del problema que resuelven.

En el marco de ATAM se toma como definición de arquitectura la realizada en [11] que establece que 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. La Arquitectura de Software condiciona las características del producto final en cuanto a cualidades o atributos de calidad como performance y mantenibilidad, por lo que resulta importante poder evaluar el cumplimiento de los mismos en forma temprana para corregir

errores antes de pasar a la codificación del sistema, donde es más costoso. Sin embargo, evaluaciones a posteriori resultan útiles como forma de aprendizaje y estudio de posibilidades de mejora, por ejemplo para desarrollar una nueva versión o para entender un sistema legado.

2.1 Conceptos en ATAM

En esta sección se describen los principales conceptos de la metodología ATAM según [5]. Para evaluar un diseño arquitectónico contra los requerimientos de atributos de calidad es necesario contar con una caracterización de los atributos de calidad. Por ejemplo para entender una arquitectura desde el punto de vista de la modificabilidad requiere entender como observar y medir este atributo, así como entender de que forma impactan en este atributo los distintos tipos de decisiones arquitectónicas. ATAM brinda una caracterización para varios atributos creada a partir del conocimiento existente en las comunidades de atributos de calidad, principalmente para atributos como performance, modificabilidad, disponibilidad, usabilidad y seguridad. Cada caracterización de atributos de calidad se divide en tres categorías: estímulo externo, decisiones arquitectónicas y respuestas, donde los primeros son eventos que hacen que la arquitectura responda o cambie, las decisiones arquitectónicas son los elementos de la arquitectura (componentes, conectores y sus propiedades) que tienen impacto directo en la obtención de respuestas a los atributos, y las respuestas se caracterizan por atributos medibles como latencia y throughput.

Como se mencionó previamente, en general no se cuenta con una especificación precisa de los requerimientos de los atributos de calidad para los sistemas, por lo que en general la primer tarea a realizar en una evaluación de arquitectura es obtener una especificación precisa de los objetivos de calidad contra los cuales la arquitectura será evaluada. El mecanismo utilizado para obtener esta información es el Escenario. Un escenario es una especificación corta que describe la interacción entre un stakeholder y el sistema, siendo la definición de stakeholder cualquier involucrado en el desarrollo, tanto cliente, como desarrolladores como gerentes, entre otros. Los escenarios proveen una forma de concretizar cualidades a tener en cuenta en tiempo de desarrollo como la modificabilidad, ayudan a entender cualidades de tiempo de ejecución como performance o disponibilidad, entre otros. Se definen tres tipos de escenarios: de Casos de Uso que involucran usos típicos del sistema y se utilizan para obtener información del mismo, similares a los definidos en la metodología de Casos de Uso que utiliza el RUP [4], escenarios de crecimiento que cubren el principio de anticipación al cambio teniendo en cuenta los posibles cambios futuros del sistema, y escenarios de exploración que cubren cambios extremos que se espera que estresen al sistema. Estos casos distintos de escenarios se utilizan para probar el sistema desde distintos puntos de vista, permitiendo que no se pasen por alto decisiones arquitectónicas que puedan representar riesgos. La información de los escenarios es obtenida y priorizada en ATAM utilizando dos mecanismos diferentes en distintos momentos, y con la participación de diferentes stakeholders, estos son los árboles de utilidad y la lluvia de ideas estructurada.

Los árboles de utilidad se realizan con el Arquitecto y proveen un mecanismo de enfoque top-down para traducir los principales requerimientos del negocio para el sistema, en escenarios concretos de atributos de calidad. Por ejemplo un requerimiento del tipo “la seguridad es central para el éxito del sistema ya que los datos de los clientes son de extrema importancia” no es un objetivo específico de calidad que pueda ser evaluado, y debe ser traducido a una especificación más concreta y específica, además de entender la importancia relativa respecto de otros atributos de calidad como la performance, para determinar el foco de la evaluación. El árbol de utilidad se arma partiendo de un nodo raíz que representa la cualidad global en estudio, y a partir de ese nodo se van detallando los requerimientos de atributos de calidad hasta llegar a expresiones concretas en las hojas del árbol, que definen los escenarios para la evaluación. Estos escenarios se priorizan en dos dimensiones: por la importancia que tiene cada uno para el éxito del sistema, y por el grado de dificultad que posee para ser realizado, según estima el Arquitecto. En ambas dimensiones se utiliza habitualmente la escala High, Medium y Low. La salida de la generación del árbol de utilidad es una lista priorizada de los requerimientos de los atributos de calidad especificados como escenarios, esto permite que el equipo de evaluación se concentre en aquellas propuestas arquitectónicas involucradas en la satisfacción de los escenarios de alta prioridad que se encuentran calificados en las hojas.

La lluvia de ideas de escenarios se realiza con todos los stakeholders involucrados en el sistema, ya que el objetivo principal es encontrar todos los escenarios posibles según la clasificación vista anteriormente. Se ha comprobado que la lluvia de ideas en grupos numerosos permite crear una atmósfera donde las ideas de una persona estimulan al resto a pensar en esa dirección y otras, promueve la comunicación, la creatividad, y sirve para expresar el pensamiento colectivo de los participantes respecto al sistema. La lista obtenida por este medio se compara con el árbol de utilidad y se agregan a éste los nuevos escenarios.

A partir del árbol de utilidad y la priorización de escenarios obtenida, se evalúa la adecuación de las propuestas arquitectónicas presentes en la Arquitectura de Software en evaluación, respecto de los atributos de calidad especificados, identificando riesgos, no riesgos, puntos de sensibilidad y puntos de concesión. Los riesgos son decisiones arquitectónicas importantes que no se han tomado en el proyecto, o que si se han tomado pero cuyas consecuencias no son comprendidas totalmente. Los no riesgos son las fortalezas de la Arquitectura, aquellas decisiones de diseño arquitectónico tomadas que son adecuadas, es decir, cumplen con los requerimientos de calidad del sistema. Los puntos de sensibilidad son parámetros en la arquitectura con los cuales se relaciona fuertemente la respuesta a algún atributo de calidad medible, por ejemplo el rendimiento o throughput afectado por la velocidad de un canal de comunicación. Un punto de concesión (o tradeoff point) se encuentra en la arquitectura cuando un parámetro arquitectónico refiere a más de un punto de sensibilidad donde los atributos de calidad medibles son afectados en forma distinta según los cambios en dicho parámetro. Por ejemplo si se aumenta la velocidad del canal de comunicación se mejora el rendimiento o throughput pero se reduce la confiabilidad, entonces es un punto de concesión. Los tres aspectos mencionados, riesgos, puntos de sensibilidad y concesión, podrían ser el foco del esfuerzo futuro de prototipado, diseño y análisis.

2.2 Metodología de ATAM

El corazón de ATAM consiste en la ejecución de nueve pasos que se dividen en cuatro grupos que su vez, se realizan en el tiempo en cuatro Fases diferenciadas. Estos cuatro grupos no se corresponden uno a uno con las Fases, sino que se realizan en las Fases 1 y 2, y se agrega una Fase 0 previa de preparación y una Fase 3 posterior de finalización del proyecto. Si bien la numeración de pasos sugiere linealidad, la ejecución de los mismos no necesariamente es un proceso en cascada estricto, ya que se podrá volver a pasos anteriores o saltar hacia adelante a pasos posteriores, o inclusive iterar entre pasos según sea necesario. Los cuatro grupos en que se dividen los nueve pasos definidos consisten en un primer grupo de presentación, donde se intercambia información del sistema, un segundo grupo de investigación y análisis, donde se valoran los atributos de calidad claves uno a uno con las propuestas arquitectónicas, un tercer grupo de pruebas donde se revisan los resultados obtenidos contra las necesidades relevantes de los stakeholders, y un cuarto y último grupo donde se presentan los resultados del ATAM.

En la Fase 0 se acuerdan tiempo, fechas, costos, esfuerzo y se forma el equipo de evaluación, en las Fases 1 y 2 se realiza la evaluación con ATAM siguiendo los nueve pasos especificados, y finalmente en la Fase 3 se realiza el informe final de la evaluación realizada, se recoge información para la medida y mejora del proceso y se actualizan los repositorios de productos generados. A continuación se describen las Fases 1 y 2 que constituyen la base de ATAM.

2.2.1 Fase 1 de ATAM

En la Fase 1 se realizan las actividades correspondientes a los grupos de presentación e investigación y análisis. El grupo de presentación se compone de tres pasos: 1 – Presentar el ATAM, 2 – Presentar las pautas del negocio y 3 – Presentar la Arquitectura. En el primer paso el equipo de ATAM presenta el método a los stakeholders explicando el proceso a seguir y el involucramiento y responsabilidad de cada uno en el proyecto. Se detallan los pasos a seguir, las técnicas a utilizar y los resultados a obtener. En el segundo paso un director de proyecto o gerente presenta el sistema desde el punto de vista del negocio, al equipo de ATAM y a los stakeholders, detallando principales funcionalidades, restricciones y metas definidas para el sistema. En el tercer paso el Arquitecto presenta la Arquitectura de Software definida incluyendo por lo menos los estilos utilizados, otros sistemas con que se debe interactuar, restricciones técnicas de software como por ejemplo uso de sistema operativo. Si se cuenta con un documento de Software Architecture Description (SAD) el Arquitecto debería basar su explicación en las distintas vistas contenidas en el mismo.

El segundo grupo de investigación y análisis se compone también de tres pasos: 4 – Identificar las propuestas arquitectónicas, 5 – Generar el árbol de utilidad de los atributos de calidad y 6 – Analizar las

propuestas arquitectónicas. En el cuarto paso el equipo de ATAM le pide al Arquitecto que identifique las propuestas arquitectónicas o estilos de arquitectura utilizados, ya que éstos definirán las estructuras importantes definidas para el sistema y las características implicadas. En el quinto paso se genera el árbol de utilidad donde principalmente el Arquitecto pero también los principales stakeholders, identifican, priorizan y refinan los requerimientos de atributos de calidad más importantes del sistema, según se mencionó previamente, identificando los escenarios en el árbol y su importancia en las dos dimensiones definidas. En el sexto paso se analizan las propuestas arquitectónicas según el árbol de utilidad generado, esto es, que tan adecuados son el uno para el otro, evaluando como cada propuesta arquitectónica influye en la obtención o no del atributo de calidad requerido, e identificando los riesgos, no riesgos, puntos de sensibilidad y concesión asociados a dicha evaluación.

2.2.2 Fase 2 de ATAM

En la Fase 2 se realizan las actividades incluidas en el tercer grupo de pruebas y el cuarto grupo de informes, que completan los tres pasos restantes. El grupo de pruebas se compone de dos pasos: 7 – Lluvia de ideas y 8 – Analizar las propuestas arquitectónicas y el grupo de Informes se compone de un solo paso: 9 – Presentar los resultados. En el séptimo paso se confirman e identifican nuevos escenarios según varios stakeholders involucrados, los que también se priorizan y se comparan con los identificados en el árbol de utilidad. Pueden suceder tres casos: el escenario ya está en el árbol de utilidad, no está pero encaja en alguna rama y se convierte en una nueva hoja, no está y no encaja en ninguna rama, lo que significa que no había sido considerado previamente. En el octavo paso se realiza lo mismo que en el sexto paso para el nuevo árbol de utilidad con todos los escenarios incluidos. Finalmente en el noveno paso el equipo de ATAM presenta los resultados a los stakeholders y entrega la documentación correspondiente a las salidas del ATAM: documento de propuestas arquitectónicas, conjunto de escenarios priorizados, conjunto de preguntas basadas en los atributos, árbol de utilidad, los riesgos descubiertos, los no riesgos documentados, los puntos de sensibilidad y de concesión encontrados, más la relación entre los riesgos encontrados y su impacto en las pautas del negocio definidas.

3 Aplicación de ATAM

La aplicación de la metodología de ATAM para evaluación de Arquitecturas de Software se realizó en el contexto de las actividades del Grupo de Ingeniería de Software (Gris) en el marco del programa de construcción y prueba de modelos de proceso de aplicación en el curso “Proyecto de Ingeniería de Software”. La pasantía realizada por dos estudiantes de quinto año de la carrera comenzó a mediados del año 2005 y tenía como objetivos el estudio y aplicación de ATAM para evaluar la herramienta de definición y generación de modelos de proceso de desarrollo de software construida por un grupo de estudiantes en el curso del año 2004.

En dicho curso un docente del Gris cumplió el rol de cliente especificando los requerimientos para el desarrollo de la herramienta, que debía permitir como mínimo la definición y generación del proceso base adaptación del RUP que se utiliza en el curso. Esta herramienta debía permitir la especificación de distintos elementos de modelado como ser fases, hitos, disciplinas, actividades, roles y entregables definidos para el modelo de proceso. Asimismo resultaba interesante la posibilidad de permitir que los entregables generados en el marco de la realización del proceso por los estudiantes, pudieran ser enviados a través de una página web y gestionados en la instanciación del proceso que se estuviera probando en esa edición del curso. El producto obtenido denominado Graphead [12] tenía como fortaleza la generación del sitio web que resultaba muy completa y amigable, pero presentaba problemas de performance en la especificación del modelo de proceso al trabajar con los elementos del mismo, y no incluía manejo de las distintas versiones que se especifican para cada año.

Dado el interés que tiene para el Grupo de Ingeniería de Software (Gris) contar con dicha herramienta para la definición de distintos modelos de proceso para cada curso, se decidió realizar un segundo ciclo de desarrollo para mejorar aspectos del prototipo obtenido y agregar nuevas funcionalidades. La pasantía de aplicación de ATAM se definió como forma de apoyar este segundo ciclo de desarrollo para que se contara también como entrada del mismo, con el informe de resultados de la evaluación de la Arquitectura de Software del primer producto.

1.1 Caso de estudio con ATAM

Para la realización de la pasantía se definieron cinco etapas que planteaban: el estudio de ATAM, el estudio de la herramienta a evaluar y su documentación, la aplicación de ATAM para evaluar la Arquitectura de Software de la herramienta, la presentación de resultados de la evaluación a los docentes del Grupo de Ingeniería de Software (Gris) y al grupo que realizaría el segundo ciclo de desarrollo, y por último el cierre de la pasantía con la entrega de la documentación final de cada etapa, cuyas primeras versiones debían entregarse al finalizar cada etapa como parte del hito definido para la misma.

La evaluación elegida fue realizar un ciclo del ATAM en su formato tardío. Esta versión del método permite entender al producto en su conjunto, y dado que se planeaba realizar un segundo ciclo de desarrollo del mismo, una evaluación tardía es especialmente indicada para entender sistemas legados. Una vez elegido el método, se tuvieron que tomar decisiones sobre como se realizaría el desarrollo del mismo. La realidad estaba dada por la ausencia del equipo de desarrollo del producto en cuestión, solo contando para el entendimiento del sistema con su documentación. El único stakeholder con el que se contaba era el cliente con el cual se podía interactuar en los pasos que así lo requirieran. Este escenario marcó el desarrollo de la evaluación, adaptando los pasos del ATAM a los insumos existentes.

En la Fase 1 se pudieron realizar los pasos 1 y 2 del grupo de presentación, y el paso 3 al no contar con el Arquitecto del producto se realizó estudiando la documentación del sistema, en particular el Software Architecture Document (SAD) y el Modelo de Diseño. Del grupo de investigación y análisis se realizaron los pasos 4, 5 y 6, este último con bastante dificultad dada la falta del Arquitecto para justificar las elecciones de diseño realizadas. De la Fase 2 el grupo de pruebas no se realizó ante la falta de stakeholders del proyecto, si se hizo el grupo de informes que fue presentado a los interesados como estaba previsto.

1.1.1 Fase 1: grupo de presentación

Se realizaron los pasos correspondientes a este grupo, comenzando por el paso 1 de presentación del ATAM que se hizo para los docentes del Grupo de Ingeniería de Software (Gris) entre los cuales se encontraba el cliente del producto en evaluación. Una de las instancias en la que la participación del cliente es clave es el paso 2 de presentación de las pautas del negocio. Durante la misma, el cliente explicó la finalidad que tiene el producto, y los aspectos de Graphhead que no le satisficieron. Del mismo modo, transmitió cuales eran sus expectativas de la evaluación, y también como esperaba que esta ayudara a mejorar la calidad del producto en la próxima iteración de mismo. Fue en esta reunión donde el equipo de evaluación obtuvo los requerimientos de los atributos de calidad que el tenía del sistema, los cuales guiarían el resto de la evaluación. Además se estudió la documentación del producto Graphhead de la disciplina Requerimientos: el Modelo de Casos de Uso, Especificación de Requerimientos, Glosario y Modelo de Dominio. Una vez conocidas las pautas del negocio, el equipo de evaluación debió enfrentarse a la Arquitectura de Software del sistema por si solo, dado que como ya fue mencionado no se contaba con la presencia del arquitecto. Por lo tanto no se realizó el paso de presentación de la Arquitectura.

1.1.2 Fase 1: grupo de investigación y análisis

Para poder identificar y entender las propuestas arquitectónicas realizadas por el Arquitecto según lo que se indica en el paso 4 de ATAM, se recurrió a la documentación existente. Se estudiaron varios documentos, principalmente el Software Architecture Document (SAD) y el Modelo de Diseño, donde en el primero se presenta la Arquitectura definida en base a los Casos de Uso identificados como relevantes. A partir de este estudio se obtuvo una visión general de la Arquitectura y de las principales decisiones tomadas, por ejemplo en lo que tiene que ver con los estilos o patrones utilizados y la descomposición en subsistemas realizada. En la figura 1 se muestra el diagrama general de la Arquitectura definida, que sigue un estilo en capas relajado, esto es, capas superiores acceden a capas inferiores no inmediatas:

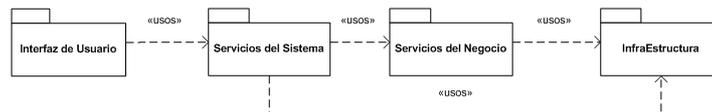


Figura 1. Estilo en capas utilizado en la Arquitectura de Graphead

La Interfaz de Usuario tiene como objetivo el manejo de la lógica del usuario y contiene el conjunto de componentes de Jgraph [13] y módulos que encapsulan la lógica de los casos de uso. Los Servicios del Sistema representan los servicios básicos que debe proveer el sistema los que son directamente utilizados por los módulos de la capa superior, constituyendo la entrada a la lógica del sistema. Los subsistemas que contiene ofrecen las interfaces requeridas por cada caso de uso asociado al subsistema y los controladores que las implementan. Los Servicios de Negocio son servicios de manejo de información del negocio, aun más básicos que los de la capa superior, lo que permite reutilizar estos módulos en otros subsistemas. En la capa de infraestructura se ubican módulos adaptadores y de servicio general, útiles para cualquiera de los subsistemas, como es el encapsulador del DBMS utilizado. Luego se describen los subsistemas definidos en cada capa y las responsabilidades asociadas a cada uno, así como los COTS utilizados en el desarrollo.

Sin embargo, la documentación de la Arquitectura debería cubrir, según ATAM, las restricciones técnicas, otros sistemas con los cuales se debe interactuar y las propuestas arquitectónicas utilizadas para alcanzar los requerimientos de los atributos de calidad. ATAM centraliza el análisis de una arquitectura en el entendimiento de sus propuestas arquitectónicas. Se concentra en identificar propuestas y estilos arquitectónicos porque estos representan la manera en que la arquitectura cumple con los atributos de calidad, esto significa que se asegura que los requerimientos críticos serán alcanzados en forma predecible. Estas propuestas arquitectónicas definen las estructuras importantes del sistema, y describen como el sistema puede crecer, responder a cambios, entre otros.

Luego de analizar la documentación existente del sistema, siguiendo las premisas propuestas por ATAM, se descubrieron distintas falencias en dicha documentación que influyen negativamente en el posterior desarrollo de la evaluación, entre las que se pueden destacar: 1) los casos de uso relevantes para la arquitectura en el SAD no se sabe porque fueron seleccionados, no existe una justificación para la elección de los mismos. 2) no se documentan los componentes que encapsulan las comunicaciones con software externo, en particular el uso del framework Hibernate [14] es oscuro. No se sabe como se maneja la actualización de datos a través del mismo, cuando según el Modelo de Diseño todas las colaboraciones entre componentes allí existentes, realizan persistencia. 3) no se especifica que componentes encapsulan el manejo del resto del software externo utilizado, según la presentación del producto, como ser Castor[15] para mapeo de clases java y archivos XML, Xalan[16] para generación de HTML a partir de XML mediante transformación XSL[17], Fop[18] para generación de PDF a partir de XML mediante transformación XSL, y Jasper[19] para generación de la agenda del proceso en formatos html, pdf y xls. 4) el Modelo de Diseño, así como el SAD, no refina los componentes críticos del sistema con la especificidad necesaria como para entender como se satisfacen los requerimientos. En particular, no se sabe como se alcanzan los requerimientos de los atributos de calidad. 5) Los requerimientos de calidad deseados por el cliente, como ser la performance, no se encuentran especificados. 6) La Documentación Técnica, es solo un listado de paquetes e interfaces. No se conoce el comportamiento de cada clase o interfaz. En vista de la presente dificultad para comprender en detalle las propuestas arquitectónicas descritas en la documentación, el equipo de evaluación gestionó una entrevista con el Arquitecto, la que finalmente pudo ser realizada. En la misma se obtuvo valiosa información que no se desprendía en forma directa de la documentación evaluada. Consideramos que tuvo una influencia positiva en el resultado de la evaluación, y confirma porque para ATAM es tan importante la participación de los distintos stakeholders, en particular el Arquitecto, en la evaluación.

El paso 5 de generación del árbol de utilidad se realizó en base a los requerimientos de los atributos de calidad obtenidos en la entrevista con el cliente. Este árbol contiene los escenarios que ponen a prueba a las propuestas arquitectónicas, para ver si estas alcanzan los requerimientos de los atributos de calidad especificados. Para realizar este estudio se generaron veintidós escenarios distintos que modelan los requerimientos de los atributos de calidad especificados por el cliente. Cada uno de estos escenarios fue ubicado en una hoja del árbol de utilidad, según el atributo de calidad que estuviera poniendo a prueba. Teniendo en cuenta la importancia de cada escenario para el éxito del sistema dada por el cliente se asignó la priorización en la primera dimensión, y según el grado de dificultad estimado por el grupo de evaluación para realizarlo, se asignó la priorización en la segunda dimensión. En la figura 2 se muestra una selección de los escenarios generados, para algunos de los atributos de calidad especificados:

Atributo de Calidad	Refinamiento del Atributo	Escenarios
Performance	Tiempo de respuesta	Informar de la aceptación de la carga de un proceso dado en menos de 0.3 segundos (M,L)
	Throughput	Versionar un proceso con 300 elementos en menos de 10 segundos (H,H)
	Latencia	Cargar un proceso con 300 elementos en menos 3 minutos (H,M)
	Carga de trabajo	Agregar un nuevo elemento a un proceso de 300 elementos, no aumentará los tiempos de latencia en más de un 15%(H,H)
Maintainability	Cambios en un subsistema no requiere cambios en otros subsistemas	Cambiar el motor de la base de datos implica cambiar un subsistema (L,M)
		Cambiar un COTS no requiere cambiar más de un subsistema (L,M)
Modifiability	Cambiar COTS	Cambiar el motor de la base de datos en menos de 10 día/persona (L,M)
		Cambiar el generador web en menos de 5 día/persona (L,M)
		Cambiar el generador pdf en menos de 5 día/persona (L,M)
Reusability	Reusabilidad de componentes	Los conceptos del negocio son utilizables por distintas aplicaciones (M,M)

Figura 2. Árbol de utilidad generado con ATAM

Los escenarios generados fueron ponderados y posteriormente priorizados por el equipo de evaluación, según los lineamientos marcados por el cliente. Una vez ordenados por la prioridad elegida, del total de los mismos se seleccionaron seis que fueron validados con el cliente, para confirmar que el esfuerzo de la evaluación estaba bien enfocado según sus expectativas. Una vez obtenidos los escenarios priorizados se pudo comenzar con la siguiente etapa de análisis de propuestas arquitectónicas, para evaluar el cumplimiento o no de los atributos de calidad especificados en los escenarios priorizados. En la figura 3 se muestran los seis escenarios priorizados:

Ranking	Escenarios	Observaciones
1	Versionar un proceso con 300 elementos en menos de 10 segundos	Performance
3	Agregar un nuevo elemento a un proceso de 300 elementos, no aumentará los tiempos de latencia en más de un 15%	Performance
4	Cargar un proceso con 300 elementos en menos 3 minutos	Performance
5	Informar de la aceptación de un alta de actividad en menos de 0.3 segundos	Performance
6	Los conceptos del negocio son utilizables por distintas aplicaciones	Reusability

Figura 3. Escenarios priorizados para la evaluación con ATAM

En el paso 6 de análisis de propuestas arquitectónicas se evaluaron diferentes propuestas arquitectónicas para los escenarios priorizados. Estas propuestas intentan satisfacer los escenarios que describen los requerimientos de los atributos de calidad. No todas las propuestas analizadas fueron realizadas por el arquitecto, sino que la mayoría fueron elaboradas por el equipo de evaluación debido a las dificultades mencionadas sobre la documentación y a pesar de la entrevista con el Arquitecto. En el análisis de cada propuesta se buscó determinar sobre que atributos de calidad influía y que riesgos o no riesgos introducía, así como los puntos de sensibilidad y de concesión identificados en cada caso. Por ejemplo para el escenario 1 – Versionar un proceso de 300 elementos en menos de 10 segundos, que corresponde al atributo de calidad general performance en su refinamiento throughput, se identificaron dos propuestas arquitectónicas relativas a la forma de acceder a los datos para realizar el versionado, cuyo análisis se resume en la figura 4:

Análisis de una Propuesta Arquitectónica					
Escenario #: 1	Escenario	Versionar un proceso con 300 elementos en menos de 10 segundos			
Atributo	Performance – Throughput				
Entorno	Proceso con 300 elementos				
Estímulo	Se invoca el versionado				
Respuesta	Versiona el proceso en menos de 10 segundos				
Decisión Arquitectónica	Sensitivity	Tradeoff	Riesgo	No riesgo	
Cada subsistema accede directamente a la BD.	S1	T1,T2	R1,R2		
Un subsistema que encapsule el acceso a datos (p.e.: uso de Hibernate)	S2	T3			
Razonamiento	S1. Disminuye la confiabilidad. S2. Aumenta la confiabilidad. R1. Aumenta el riesgo que los datos queden inconsistentes ante una eventual falla. R2. Aumento de dependencia entre la aplicación y la BD. T1. Mejora la performance, pero disminuye la maintainability. T2. Mejora la performance, pero disminuye la reusability. T3. Aumenta la reusability, pero empeora la performance.				

Figura 4. Análisis de propuestas arquitectónicas del escenario 1

En el primer caso se accede directamente desde cada subsistema y en el segundo caso se encapsula el acceso en un subsistema al que accede el resto. Para el primer caso se identificaron dos riesgos, uno la posibilidad de que los datos quedaran inconsistentes ante una eventual falla, y otro que se aumentaba la dependencia del sistema a la base de datos elegida. Como puntos de concesión del atributo performance evaluado se vio que empleando esta opción se mejoraba la performance pero disminuían la mantenibilidad y la reusabilidad, ya que por ejemplo, cambiar de manejador de base de datos requeriría un esfuerzo importante. En el segundo caso no se identificaron riesgos, como punto de concesión se evaluó que se incrementaba la reusabilidad y mantenibilidad, pero se disminuía la performance. Se identificó como punto de sensibilidad para las dos propuestas evaluadas, la confiabilidad, que en la primer opción disminuye y en la segunda aumenta.

Además de la evaluación de las propuestas arquitectónicas frente a los escenarios priorizados, se obtuvieron resultados adicionales debido al estudio de la documentación y de la entrevista con el Arquitecto en la cual se conocieron detalles de la implementación, donde según él radicaban la mayoría de los problemas asociados a la degradación de performance detectada. Estos resultados fueron también de utilidad para apoyar al grupo realizando el segundo ciclo de desarrollo, destacando los relacionados a problemas en el acceso a datos mediante Hibernate por un uso no del todo adecuado del framework, la decisión de actualizar la base de datos en tiempo real en lugar de aprovechar el caché mientras se está trabajando con un proceso y persistir al final del mismo, y la utilización de algunos algoritmos que no alcanzaban tiempos de ejecución adecuados. Por otro lado, se rescataba como de gran utilidad para el desarrollo el uso de otros componentes COTS como ser Castor, Xalan, Fop, Jasper y Jgraph.

1.1.3 Fase 2: grupo de pruebas

Debido a la realidad a la cual se enfrentaba el equipo de evaluación, se omitió la instancia en la cual todos los stakeholders proponen escenarios, y luego los analizan junto con los evaluadores. ATAM considera que a pesar de que los evaluadores a esta altura llevan un buen tiempo trabajando sobre la arquitectura y la conocen en detalle, siempre quedan algunas características ocultas, las cuales pueden llegar a ser críticas para alcanzar los requerimientos de los atributos de calidad. Es por este motivo que se introduce este paso. Por lo tanto, al no realizarlo, se perdió la oportunidad de descubrir nuevos puntos de sensibilidad, de concesión, riesgos y no riesgos que pudieran haberse pasado por alto.

1.1.4 Fase 2: grupo de pruebas

Para este paso se debió realizar nuevamente una adaptación sobre lo que se indica en ATAM, ya que se debían presentar los resultados a dos grupos distintos, por un lado el Grupo de Ingeniería de Software (Gris) interesado tanto en la evaluación del producto Graphead como en la metodología ATAM y las adaptaciones realizadas a la misma en el marco de la pasantía. Por otro lado, al grupo de estudiantes que iba a realizar el segundo ciclo de desarrollo sobre el producto, a quienes interesaban principalmente las guías que se pudieran dar sobre la Arquitectura de Software evaluada y los aspectos a tener en cuenta según los riesgos, no riesgos, puntos de concesión y de sensibilidad encontrados, así como las posibilidades de reuso existentes, no siendo factible entrar en detalles de la metodología ATAM aplicada. Por lo tanto, se hicieron dos presentaciones, una como indica ATAM entrando en los detalles de la evaluación realizada, y otra orientada a la conceptualización de los resultados obtenidos en la evaluación. Las principales observaciones fueron las correspondientes a los problemas detectados en el manejo de la persistencia, recomendando realizar nuevamente este desarrollo, y el de adecuación del resto de los COTS utilizados, en particular el generador del sitio web y documento pdf del proceso, recomendando la reutilización de dicho componente y los COTS que éste incluía.

2 Conclusiones y trabajo futuro

Como resultado de la aplicación del método se obtuvo una lista de escenarios priorizados y las diferentes propuestas arquitectónicas que los intentan satisfacer, a partir de las cuales se descubrieron puntos de sensibilidad, de concesión, y riesgos asociados. Estas propuestas arquitectónicas son una excelente guía para tomar futuras decisiones arquitectónicas, ya que se analizó su impacto en la Arquitectura, y atacan directamente las expectativas de calidad que el cliente tiene del sistema.

Consideramos que el método resultó muy útil para evaluar la Arquitectura de Software, a pesar de que no se siguió la metodología propuesta en forma estricta, incluso aún omitiendo y modificando pasos según las posibilidades con que se contaba. La evaluación permitió un enfoque amplio del sistema, en lugar de uno estrecho o con miras a corto plazo. Gracias a este tipo de enfoque que plantea el método de evaluación se descubrieron riesgos ocultos y errores que explican los problemas que tiene el producto. Creemos que ATAM se puede adaptar a las necesidades particulares de cada negocio, como se hizo en este caso. El segundo ciclo de desarrollo fue exitoso y se pudieron mejorar los problemas de performance del producto, manteniendo los aspectos positivos detectados reutilizando los componentes mencionados. Una descripción de este producto puede verse en [20].

Como trabajo a futuro se piensa incluir una adaptación de método ATAM en las actividades de la Disciplina de Aseguramiento de la Calidad del proceso base adaptación del RUP del curso “Proyecto de Ingeniería de Software”, que permitan realizar una evaluación de la Arquitectura de Software definida antes de que sea implementada. Creemos que realizar esta evaluación permitirá encontrar puntos de sensibilidad, de concesión, de riesgos y no riesgos para las propuestas identificadas que aportarán a la mitigación de riesgos en las decisiones de diseño tomadas, antes de la implementación de la Línea Base de la Arquitectura en la Fase de Elaboración.

Luego del estudio realizado por la pasantía, el método ATAM se incluyó en el temario de la asignatura electiva de quinto año de la carrera “Taller de Arquitectura de Software”[21] dictada por el Grupo de Ingeniería de Software (Gris) en el marco de las áreas de investigación del grupo.

3 Referencias

- [1] Grupo de Ingeniería de Software (Gris), Instituto de computación, Facultad de Ingeniería, Universidad de la República, <<http://www.fing.edu.uy/inco/cursos/ingsoft/gris>> [Consulta: diciembre de 2006]
- [2] Proyecto de Ingeniería de Software, Instituto de computación, Facultad de Ingeniería, Universidad de la República, <<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/>> [Consulta: diciembre de 2006]
- [3] Delgado, A. Pérez, B. “Modelo de Desarrollo de Software OO – Experimentación en un curso de Ingeniería de Software”, en V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC’06), Puebla, México, Febrero de 2006, ISBN 970-94770-0-5
- [4] IBM Rational Unified Process. <<http://www-130.ibm.com/developerworks/rational/products/rup/>> [Consulta: diciembre de 2006]
- [5] Clements P., Kazman R., Klein R., Evaluating Software Architectures, Methods and Case Studies, Addison-Wesley Pearson Education, 2004, ISBN 0-201-70482-X
- [6] Software Engineering Institute <<http://www.sei.cmu.edu/>> [Consulta: diciembre de 2006]
- [7] Castro A., Germán M., Pasantía de ATAM <<http://www.fing.edu.uy/inco/cursos/ingsoft/atam.htm>> [Consulta: diciembre de 2006]
- [8] Software Architecture Analysis Method (SAAM) <<http://www.sei.cmu.edu/publications/articles/saam-method-property-sas.html>> [Consulta: diciembre de 2006]
- [9] Quality Attribute Workshop (QAW), <<http://www.sei.cmu.edu/architecture/qaw.html>> [Consulta: diciembre de 2006]
- [10] Shaw M., Garlan D., Software Architecture: perspectives on an emerging discipline, Prentice-Hall, 1996, ISBN 0-131-82957-2
- [11] Bass L., Clements P., Kazman R., Software Architecture in practice, Addison-Wesley, 2003, ISBN 0-321-15495-9
- [12] Proyecto de Ingeniería de Software, Memoria Organizacional, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Grupo 3 año 2004, <<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/>> [Consulta: diciembre de 2006]
- [13] Jgraph, <<http://sourceforge.net/projects/jgraph/>> [Consulta: diciembre de 2006]
- [14] Hibernate, <<http://www.hibernate.org/>> [Consulta: diciembre de 2006]
- [15] Castor, <<http://www.castor.org/>> [Consulta: diciembre de 2006]
- [16] Xalan, <<http://xalan.apache.org/>> [Consulta: diciembre de 2006]
- [17] XSL, <<http://www.w3.org/TR/xsl/>> [Consulta: diciembre de 2006]
- [18] Fop, <<http://xmlgraphics.apache.org/fop/>> [Consulta: diciembre de 2006]
- [19] Jasper, <<http://jasperforge.org/sf/projects/jasperreports>> [Consulta: diciembre de 2006]
- [20] Pérez, B., DeVeloPro, Herramienta para la Documentación y Gestión de los Procesos de una Organización, CACIC 2006, San Luis, Argentina.
- [21] Taller de Arquitectura de Software, Instituto de computación, Facultad de Ingeniería, Universidad de la República, <<http://www.fing.edu.uy/inco/cursos/tarqsoft>> [Consulta: diciembre de 2006]

Agradecimientos

El presente trabajo ha sido desarrollado en el marco del proyecto COMPETISOFT (Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria de Software de Iberoamérica) del programa CYTED (Ciencia y Tecnología para el Desarrollo).