

Evaluación Arquitectónica de un Software Basado en Componentes

Maria Pérez¹, Luis Mendoza¹, Dakar Parada², George Di Paula²

¹ Laboratorio de Investigación en Sistemas de Información
Departamento de Procesos y Sistemas
Universidad Simón Bolívar
Caracas - Venezuela

{movalles, lmendoza}@usb.ve
² Coordinación Ingeniería de Sistemas
Universidad Nacional Experimental Politécnica
de la Fuerza Armada Nacional –UNEFA-
Caracas - Venezuela
dakarpp@yahoo.com, geopower@hotmail.com

Resumen. El enfoque de desarrollo de software en componentes constituye una novedosa manera de construir aplicaciones. En este sentido, el desarrollo de componentes propicia características de calidad en el producto de software que otros enfoques de desarrollo no impulsan. Sin duda una arquitectura de calidad propicia características de calidad en el sistema resultante. En este trabajo se evaluó la arquitectura de un software basado en componentes sobre la plataforma J2EE aplicando el método ATAM, el objetivo principal fue verificar que el desarrollo del mismo terminase en un producto que cumpliera con los requerimientos del cliente, el logro de las metas del negocio y la calidad total del software. Su principal aporte es que permitió constatar que las características de calidad que propicia una arquitectura de software basada en componentes, son la Portabilidad y Mantenibilidad.

1 Introducción

Con la madurez que alcanza lentamente el enfoque de componentes, viene una desvinculación a igual velocidad del enfoque orientado a objetos. Sin embargo, mucho se puede aprender de tecnología de objetos, y algo de ella se puede generalizar o transformar para servir a la de componentes [3]. En el mundo de componentes, los puntos prioritarios giran alrededor de conceptos arquitectónicos.

Por su parte, el estudio de la arquitectura de software sólo tiene sentido si las decisiones que se tomen sobre esa arquitectura determinan las características de calidad [2] del sistema; en consecuencia, es conveniente evaluar las decisiones de tipo arquitectónico con respecto al impacto que ellas causarán sobre esas características. La arquitectura de software también hace posible determinar la estructura del proyecto de desarrollo del sistema [1]. De lo anterior se puede afirmar que el establecimiento de

una arquitectura correcta constituye un factor fundamental en el éxito del proceso de desarrollo del sistema y el cumplimiento de las características de calidad.

En este trabajo se evalúa la arquitectura de un software basado en componentes aplicando el método ATAM [1]; esta evaluación permitió identificar las características de calidad de una arquitectura basada en componentes.

2 Aplicación de ATAM

A continuación se muestran los resultados de haber aplicado los pasos de ATAM [1] al Estudio de Caso [4]. Por razones de espacio, sólo se muestra los de mayor relevancia para este artículo.

Fase 1: Presentación

- **Paso 2. Presentación del manejo del negocio.** En este paso se clarificaron las metas de negocio que motivaban el desarrollo del proyecto y los objetivos que tenían que ver con la arquitectura; resultando de mayor relevancia, las características de calidad que se especifica en la Tabla 1.

Tabla 1. Árbol de Utilidad.

| Escenarios |
|--|
| FI1. FONCRE efectúa monitoreo al SPI para saber de sus movimientos de crédito (A,A) FI2. SPI notifica a FONCRE que ha realizado los movimientos correspondientes a crédito (B,A) |
| FS1. Se trata de producir intrusión no autorizada al sistema y este la rechaza (A,A) FS2. Un usuario (no administrador) trata de ingresar al área de administración de crédito, el sistema emite el mensaje de “acceso denegado” (A,A) |
| ED1. Agregar campos a la tabla “Solicitud” hasta duplicar su tamaño y mantener el tiempo de respuesta. (M,B). ED2. Solicitud de la tabla de amortización vía Web y recibe la respuesta en menos de 5 seg. (A,B) ED3. Un acreditado solicita el estado de su cuenta vía Web y recibe la respuesta en menos de 5 seg. (A,B) |
| MM1. Una modificación en la tabla de estatus para que guarde la información de los evaluadores de la solicitud, en dicho estatus se realiza en menos de una semana de trabajo de una persona (B,M) MM2. Agregar un nuevo tipo de tabla de amortización en menos de una semana (A,B) MM3. Se incrementan los tipos de solicitud para otro tipo de servicios, lo cual requiere efectuar un mantenimiento al sistema (configuración). Dicho mantenimiento se completa en menos de una semana (B,A) |
| PA1. Reemplazo del sistema operativo, y no tener que realizar cambios en el código (A,B) PA2. El sistema FONCRE se instala en Brasil, debe cambiarse el idioma. (A,A) PA3. Cambio del contenedor de aplicaciones OC4J a JBOSS (A,B) |
| PC1. Se crea una solicitud de crédito y FONCRE busca los datos del solicitante en la base de datos de SPI (A,A) |
| <u>Legenda:</u> FI: Funcionalidad-Interoperabilidad, FS: Funcionalidad-Seguridad, ED: Eficiencia-Desempeno, MM: Mantenibilidad-Modificabilidad, PA: Portabilidad-Adaptabilidad, PC: Portabilidad-Coexistencia. |

Fase 2: Investigación y Análisis

- **Paso 5. Árbol de Utilidad.** Para cada escenario se hizo una priorización en base a dos dimensiones: 1) importancia del escenario en relación al éxito del sistema y 2) grado de dificultad para el logro del escenario. Esta priorización se hizo con la escala: alto (A), medio (M) y bajo (B) (ver Tabla 1).
- **Paso 6: Análisis de los escenarios.** En este paso se analizaron los escenarios más importantes de acuerdo al árbol de utilidad, estableciendo en cada uno de ellos las posibles decisiones arquitectónicas a tomar, y para cada decisión sus puntos de sensibilidad (cuando la decisión afecta sólo una característica de calidad), sus riesgos, no riesgos y *tradeoff* (cuando la decisión afecta más de una característica de calidad). En la Tabla 2 se presenta el escenario FI1.
- **Paso 9: Presentación de resultados.** Las características de calidad propiciadas a través del conjunto de decisiones arquitectónicas que se tomaron, fueron, en primer lugar la Portabilidad mediante la Adaptabilidad y, en segundo lugar, la Mantenibilidad mediante la Modificabilidad. Igualmente se propició la Funcionalidad a través de las decisiones relativas a la Seguridad. Para el escenario FI1 se decidió seleccionar la decisión arquitectónica número dos debido a que es una meta primordial del negocio el hecho de mantener la independencia entre el FONCRE y el sistema SPI.

4 Conclusiones

Se realizó la evaluación arquitectónica de un software basado en componentes. De dicha evaluación se puede decir que la característica de calidad más deseada en una arquitectura de componentes, es la Mantenibilidad. De la misma manera, se determinó que las características de calidad que una arquitectura de componentes propicia son: Mantenibilidad y Portabilidad

Se pudo observar que la utilización de ATAM es adecuada para la evaluación de este tipo de arquitecturas, ya que el método permite una evaluación temprana; es decir, no es necesario que toda la arquitectura este definida para ser evaluada; el comportamiento observado en el alcance de la arquitectura evaluada se propaga a la arquitectura completa. Por esta razón se afirma que el comportamiento conseguido en la evaluación es válido para futuros componentes que se adhieran a la arquitectura de software, si ellos cumplen con los mismos patrones que los ya evaluados.

Tabla 2. Escenario FI1.

| | | | | |
|---|--|-------------------------------|-----------------|-------------------|
| Código Escenario: FI1 | FONCRE efectúa monitoreo al SPI para constatar si se han realizado los movimientos de crédito. | | | |
| Característica-Subcaracterística | Funcionalidad-Interoperabilidad | | | |
| Estímulo | Operaciones normales | | | |
| Respuesta | Se efectúa el Monitoreo al SPI | | | |
| Decisión arquitectónica | Riesgos | Puntos de Sensibilidad | Tradeoff | No riesgos |

| | | | | |
|---|----------|----|----|----|
| DA 1 Lectura directa a la BD de SPI | R1 | S1 | | |
| DA 2 Lectura de un archivo generado por SPI | R2,R3,R4 | S2 | | |
| DA 3 Componente en SPI que escriba en la BD de FONCRE | | S3 | T1 | N1 |
| Razonamiento | | | | |
| R1 Se produce una dependencia, entre FONCRE y SPI en la estructura de los datos. | | | | |
| R2 Posibilidad de error en la lectura del archivo. | | | | |
| R3 No ubicación del archivo. | | | | |
| R4 Corrupción de los datos contenidos en el archivo, ocasionada por agentes externos. | | | | |
| S1 La lectura directa desde FONCRE a SPI propiciará mejoras en los Tiempos de respuesta. | | | | |
| S2 Se aumenta la Confiabilidad ya que se puede leer el archivo las veces que sea necesario. | | | | |
| S3 Disminución de los niveles de Seguridad de FONCRE. | | | | |
| T1 Disminuye la Confiabilidad ya que a la hora de que falle la transmisión de datos desde el SPI, no habrá forma de recuperarse de esa falla, pero se mejora la Eficiencia. | | | | |
| N1 Representa el hecho de que la decisión arquitectónica no tiene asociada ningún riesgo. | | | | |
| Diagrama de la Arquitectura | | | | |
| <p>El diagrama de arquitectura muestra los siguientes componentes y sus relaciones:</p> <ul style="list-style-type: none"> SPI (Software Product Interface) interactúa con FONCRE (FONCRE) y Archivo (Archivo). FONCRE interactúa con Archivo y Proceso (Proceso). Archivo interactúa con Proceso. Componente externo SPI interactúa con FONCRE. Se muestran tres bases de datos (cilindros) que interactúan con FONCRE y SPI. <p>Las decisiones de arquitectura (DA) se representan en burbujas de diálogo:</p> <ul style="list-style-type: none"> DA 1 apunta a la conexión entre SPI y FONCRE. DA 2 apunta a la conexión entre FONCRE y Archivo. DA 3 apunta a la conexión entre Componente externo SPI y FONCRE. | | | | |

Referencias

1. Kazman, R., Clements, P., Klein, M.: Evaluating Software Architectures: Methods and Case Studies. Addison-Wesley. USA. (2002).
2. ISO/IEC FCD 9126-1.2: Information Technology – Software Product Quality. Part 1: Quality Model. International Standardization Organization. 1998.
3. Szypersky, C. Components Software Beyond Object-Oriented Programming. Second edition. Addison-Wesley. USA. 2002.
4. Di Paula, G., Parada, D., Evaluación Arquitectónica de un Software Basado en Componentes. TEG presentado a la UNEFA, como requisito para optar al Título de Ingeniero de Sistemas. Caracas-Venezuela, Julio 2004.