

GestioPolis^{com}
CONOCIMIENTO EN NEGOCIOS

Universidad de las Ciencias Informáticas

Torrens, la Lisa, Ciudad de la Habana, Cuba



Procedimiento para la Evaluación de Arquitecturas de Software basadas en Componentes

Artículo para la publicación de investigaciones concluidas

08

DATOS DE CONTACTOS

Datos personales

- Nombre y apellidos: *Yoan Arlet Carrascoso Puebla*
- Fecha de Nacimiento: 7 de diciembre de 1984
- Dirección: calle 1ra A e/. 14 y 16 No.1407, Reparto Costa Azul, Boca de Camarioca.
- Dirección e-mail: yacarrascoso@uci.cu
- Localidad: Varadero, Matanzas. Cuba
- Titulación: Ingeniero en Ciencias Informáticas.



Actividad profesional

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el 2008. Ha trabajado como desarrollador en sistema de gestión hotelera para el ministerio del Turismo Cubano y como arquitecto principal en Sistemas de Gestión de Inventario y Almacenes para el Ministerio del Turismo basados sobre la plataforma J2EE. Ha formado parte del equipo central de arquitectura como arquitecto de integración del proyecto ERP Cubano. Posteriormente se he desempeñado como arquitecto de sistema y de datos de la línea contabilidad financiera del proyecto ERP Cubano. Ha asumido el rol de jefe de la línea de contabilidad financiera y actualmente se desempeña como arquitecto principal de la línea de finanzas del ERP cubano que se encuentra en desarrollo basado en tecnologías libres como PHP y PostgreSQL.

Datos personales

- Nombre y apellidos: *Enrique Chaviano Gómez*
- Fecha de Nacimiento: 30 de abril de 1984
- Dirección: calle 1ra e/ E y Final #248, Rpto Libertad. Santa Clara. Villa Clara
- Dirección e-mail: echaviano@uci.cu
- Localidad: Santa Clara, Villa Clara. Cuba
- Titulación: Ingeniero en Ciencias Informáticas.



Actividad profesional

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el 2008. Ha trabajado como desarrollador en sistema de gestión hotelera para el ministerio del

Turismo Cubano basados en la plataforma J2EE, como arquitecto principal en Sistema de Gestión de Inventario y Almacenes para el Ministerio del Turismo basados sobre la plataforma J2EE, como arquitecto de Sistema del proyecto ERP (*Enterprise Resource Planning*) cubano, como arquitecto principal del Subsistema Costos y Procesos del Proyecto ERP cubano y como Jefe de Línea del Subsistema Costos y Procesos del ERP cubano, proyecto para obtener un software de gestión, del tipo ERP que se encuentra en desarrollo sobre tecnologías libres como PHP y PostgreSQL.

Datos personales

- Nombre y apellidos: *Anisleydi Céspedes Vega*
- Fecha de Nacimiento: 8 de Febrero del 1985
- Dirección: calle 53, e: e y cn, edificio 5008, apto 2, Micro 70.
- Dirección e-mail: acespedes@uci.cu
- Localidad: Nueva Gerona, Isla de la Juventud. Cuba.
- Titulación: Ingeniera en Ciencias Informáticas.



Actividad profesional

Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el 2008. Ha trabajado como planificadora en el sistema de gestión hotelera para el ministerio del Turismo Cubano y en el Sistema de Gestión de Inventario y Almacenes para el Ministerio del Turismo basados sobre la plataforma J2EE. Ha formado parte del equipo central de arquitectura del proyecto ERP Cubano también como planificadora. Posteriormente se ha desempeñado como analista principal del módulo de Cobros y Pagos de la línea contabilidad financiera del proyecto ERP Cubano. Actualmente se desempeña como jefa del módulo de Cobros y Pagos de la línea de contabilidad financiera del ERP cubano que se encuentra en desarrollo basado en tecnologías libres como PHP y PostgreSQL.

ABSTRACT

This article presents a small study of the art of the Evaluation of Software Architectures with the objective of providing an overview on the topic. In the same it treats topics such

as: What is an appraisal? Which are the objectives of assessing an architecture? Why evaluate an architecture? When make an assessment to an Architecture? Which are its characteristics? Who participates in an evaluation of an architecture? As well as techniques used in the evaluation to an architecture, a brief analysis of some of the assessment's methods and a proposal of LISI's evaluation (Research Laboratory in Information Systems), Simon Bolivar University.

RESUMEN

En el presente trabajo se realiza un pequeño estudio del estado del arte de la Evaluación de Arquitecturas de Software con el objetivo de brindar un panorama general sobre el tema. En el mismo se tratan temas como: ¿Qué es una evaluación? ¿Cuáles son los objetivos de evaluar una arquitectura? ¿Por qué evaluar una Arquitectura? ¿Cuándo realizar una evaluación a una Arquitectura? ¿Cuáles son sus características? ¿Quiénes participan en una evaluación de una Arquitectura?, así como técnicas empleadas a la hora de realizarle una evaluación a una Arquitectura, un breve análisis de algunos de los métodos de evaluación existentes y una propuesta de evaluación del LISI (Laboratorio de Investigación en Sistemas de Información), Universidad Simón Bolívar.

ÍNDICE DE CONTENIDO

DATOS DE CONTACTOS.....	3
ABSTRACT	4
RESUMEN.....	5
ÍNDICE DE CONTENIDO.....	6
ÍNDICE DE TABLAS	7
ÍNDICE DE FIGURAS.....	7
INTRODUCCIÓN.....	9
MARCO CONCEPTUAL.....	11
CAPÍTULO 1. Estudio del estado del arte.....	13
Introducción	13
Objetivos	13
¿Cómo determinamos que forma parte de una Arquitectura?	13
Principales problemas en un proyecto causados por la Arquitectura	13
¿Qué es una Evaluación?	13
Objetivos de Evaluar una Arquitecturas	14
Características de una Evaluación de Arquitectura.....	15
¿Cómo puedo estar seguro que la elección de mi arquitectura es la correcta para mi software?.....	15
¿Por qué evaluar una Arquitectura?.....	16
¿Cuándo una Arquitectura puede ser evaluada?	16
¿Quiénes participan en una Evaluación?	17
Técnicas de Evaluación	17
Planear o no una arquitectura	18
¿Por qué cualidades puede ser evaluada una Arquitectura?	18
¿Cuáles son los beneficios de realizar una evaluación arquitectónica?.....	19
¿Qué resultados produce la evaluación de una Arquitectura?.....	20
¿Cuáles son las salidas de una evaluación arquitectónica?	20
Pasos Básicos para una Evaluación	20
Métodos de Evaluación de Arquitecturas	21
CONCLUSIONES PARCIALES	28
CAPÍTULO 2. Resultados y Discusión	28

Introducción	28
Objetivos	29
Propuesta de Procedimiento	29
Equipo de Evaluación.....	29
Instrumentos y técnicas de evaluación.....	30
Fases del Método.....	30
CONCLUSIONES PARCIALES	39
CONCLUSIONES	40
RECOMENDACIONES.....	41
BIBLIOGRAFÍA	42

ÍNDICE DE TABLAS

Tabla #1. Descripción de atributos de calidad observables vía ejecución (2)	18
Tabla #2. Descripción de atributos de calidad no observables vía ejecución (2)	19
Tabla #3. Comparación entre Métodos de Evaluación (2).	22
Tabla #4. Comparación entre los métodos ALMA, PASA, SALUTA y SNA (9).	28
Tabl a#5. Grupos participantes en el método MECABIC (1).	29
Tabla #6. Subconjunto del árbol de utilidad inicial propuesto por el método (1).....	33
Tabla #7. Algunas preguntas para analizar los elementos de diseño identificados (1)	36

ÍNDICE DE FIGURAS

Figura #1: Clasificación de las Técnicas de Evaluación (4).....	18
Figura #2. Flujo conceptual del método ATAM (7)	21
Figura #3. Método ALMA (9).....	24
Figura #4. Método PASA (9)	25
Figura #5. Método SALUTA (9).	26

INTRODUCCIÓN

La Arquitectura de Software es una práctica joven de apenas unos pocos años de trabajo constante. Tiene sus orígenes en la década de 1960, pero no fue hasta los años 90 que Dewayne Perry y Alexander Wolf, la exponen en el sentido que hoy se conoce. Esta década se consideró como la década de la arquitectura de software, según profetizaron Perry y Wolf. A partir de este momento la Arquitectura de Software comenzó a tener auge vertiginoso tanto en la academia como en la industria, desarrollándose los de estilos arquitectónicos, los ADL (*Leguajes de Descripción Arquitectónica*), aunque esto es algo que aún hoy está un poco virgen, entre otros. Todo esto dando al traste con la creación de diferentes tipos de arquitecturas como las basadas en componentes.

La Arquitectura de Software de los Sistemas de Software a ser construidos, se convierte en un factor de importancia para lograr que éste tenga un alto nivel de calidad. Recuérdese que el poseer una buena Arquitectura de Software es de suma importancia, ya que ésta es el corazón de todo sistema informático y determina cuáles serán los niveles de calidad asociados al sistema (1).

No sirve de nada un sistema que no cumple con los atributos de calidad que se especificaron en los requerimientos no funcionales de los clientes. Por lo que diseñar una correcta arquitectura va a determinar el éxito o fracaso de un sistema de software, en la medida que esta cumpla o no con sus objetivos (4). Debido a esto “Para reducir tales riesgos, y como buena práctica de ingeniería, es recomendable realizar evaluaciones a la arquitectura” (4).

En este artículo se explica el procedimiento para realizar Pruebas de Concepto a una Arquitectura de Software (Evaluar una Arquitectura), enfatizando en el por qué se hace necesaria, los beneficios que brinda, así como destacar algunos métodos de evaluación existentes y cuál de ellos se propone. En este caso, el **MECABIC** (*Método de Evaluación de la Calidad de Arquitecturas Basadas en la Integración de Componentes*), cuyo objetivo principal es evaluar y analizar la calidad de este tipo de Arquitectura de Software. El método es propuesto por Aleksander González, Marizé

Mijares, Luis E. Mendoza, Anna Grimán, María Pérez, LISI (Laboratorio de Investigación en Sistemas de Información), Universidad Simón Bolívar.

EL **objetivo general** que se persigue con la elaboración de este artículo es proponer un procedimiento para evaluar Arquitecturas de Software Orientadas a Servicios. Materializándolo en la arquitectura del ERP.

Los **objetivos específicos** que se persiguen con este artículo son:

- Realizar un estudio del estado del arte de la Evaluación de Arquitecturas de Software.
- Analizar e identificar potencialidades de los diferentes métodos o procedimientos que existen para evaluar Arquitecturas de Software.
- Proponer un procedimiento de evaluación que permita constatar el grado de cumplimiento de una arquitectura con los requisitos no funcionales de un sistema.

El artículo está estructurado en dos capítulos:

En el **capítulo 1** se realiza un estudio del estado del arte de las Pruebas de Concepto brindando una visión general de los aspectos teóricos relacionados con el tema; también se detallan algunos métodos o procedimientos, sus características una breve comparación entre ellos.

En el **capítulo 2** se describe una propuesta de método de evaluación de arquitectura enfatizando en aspectos como instrumentos y técnicas de evaluación, participantes y los pasos de cada fase para el desarrollo del mismo.

El **alcance** que se persigue con este documento es dar a conocer un panorama general sobre evaluación de arquitecturas y proponer que el método propuesto se aplique a la arquitectura de los módulos del ERP (Enterprise Resources Planning), Sistema de Planificación de recursos que está desarrollando la Facultad 3 en conjunto con otras facultades de la Universidad de las Ciencias Informáticas.

MARCO CONCEPTUAL

Arquitectura de Software: Según el estándar IEEE (*Institute of Electrical and Electronics Engineers*): “*La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución*” (1).

Calidad de Software: La Calidad de Software es “*la concordancia con los requisitos funcionales y de rendimiento establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado de forma profesional*” (1).

Componente: “*Es una parte de una arquitectura de software claramente identificable, independiente a la aplicación en la que se utiliza y de otros componentes (partes), que describe y realiza funciones específicas y claras dentro del contexto de la arquitectura, puede ser modificado durante el diseño, posee una documentación clara que permite conocer sus características, atributos y comportamiento, reutilizable y su interoperabilidad con otros componentes (partes) no reduce el nivel de eficiencia de la arquitectura*” (1).

Escenario: Un escenario consta de tres partes: el estímulo, el contexto y la respuesta. El estímulo es la parte del escenario que explica o describe lo que el involucrado en el desarrollo hace para iniciar la interacción con el sistema. Puede incluir ejecución de tareas, cambios en el sistema, ejecución de pruebas, configuración, etc. El contexto describe qué sucede en el sistema al momento del estímulo. La respuesta describe, a través de la arquitectura, cómo debería responder el sistema ante el estímulo. Este último elemento es el que permite establecer cuál es el atributo de calidad asociado (2).

“*Los escenarios describen una utilización anticipada o deseada del sistema, y típicamente se expresan en una frase*” (3). “*Los mismos representan una abstracción de los requerimientos más importantes de un sistema*” (3).

Stakeholders: Aquellas personas que están relacionadas, de cierta forma, con el sistema, ya sea un desarrollador, usuario o gerente, etcétera.

CAPÍTULO 1. Estudio del estado del arte

Introducción

En el presente capítulo se realiza un estudio del estado del arte de la evaluación de arquitecturas de software. Para ir introduciéndose en el tema se tratan aspectos como los principales problemas que causa la arquitectura en un proyecto, también se define que es la evaluación de arquitecturas, sus características, objetivos que persigue. Se hace además un análisis de cuándo y por qué una determinada arquitectura debe ser evaluada y los beneficios que reporta.

Objetivos

- Brindar una información acerca de la importancia y las implicaciones que se desprenden de una evaluación arquitectónica.
- Analizar e identificar potencialidades de los diferentes métodos o procedimientos que existen para evaluar Arquitecturas de Software.

¿Cómo determinamos que forma parte de una Arquitectura?

Debe ser un componente (elemento), relación entre componentes, o una propiedad 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 (2).

Principales problemas en un proyecto causados por la Arquitectura (5):

- Este falla en cumplir con la calidad necesaria.
- Este falla en intentar soportar las necesidades de negocio.
 - Falta de compromiso del usuario para que el proyecto termine (problemas en la comunicación).
- El 50% de proyectos que se atrasan tienen problemas en la arquitectura.
- El 35% de los proyectos que exceden el costo de construcción tienen problemas en la arquitectura.

¿Qué es una Evaluación?

- Es un estudio de factibilidad que pretende detectar posibles riesgos, así como buscar recomendaciones para contenerlos (5).

- La diferencia entre evaluar y verificar es que la evaluación se realiza antes de la implementación de la solución. La verificación es con el producto ya construido (5).

Objetivos de Evaluar una Arquitecturas

El **objetivo** de evaluar una arquitectura es saber si puede habilitar los requerimientos, atributos de calidad y restricciones para asegurar que el sistema a ser construido cumple con las necesidades de los stakeholders (5).

La evaluación de una arquitectura de software es una tarea no trivial, puesto que se pretende medir propiedades del sistema en base a especificaciones abstractas, como por ejemplo los diseños arquitectónicos. Por ello, la intención es más bien **la evaluación del potencial de la arquitectura diseñada para alcanzar los atributos de calidad requeridos**. Las mediciones que se realizan sobre una arquitectura de software pueden tener distintos objetivos, dependiendo de la situación en la que se encuentre el arquitecto y la aplicabilidad de las técnicas que emplea. Algunos de estos objetivos son: *cualitativos, cuantitativos y máximos y mínimos teóricos* (2).

- ✓ La **medición cualitativa** se aplica para la comparación entre arquitecturas candidatas y tiene relación con la intención de saber la opción que se adapta mejor a cierto atributo de calidad. Este tipo de medición brinda respuestas afirmativas o negativas, sin mayor nivel de detalle.
- ✓ La **medición cuantitativa** busca la obtención de valores que permitan tomar decisiones en cuanto a los atributos de calidad de una arquitectura de software. El esquema general es la comparación con márgenes establecidos, como lo es el caso de los requerimientos de desempeño, para establecer el grado de cumplimiento de una arquitectura candidata, o tomar decisiones sobre ella. Este enfoque permite establecer comparaciones, pero se ve limitado en tanto no se conozcan los valores teóricos máximos y mínimos de las mediciones con las que se realiza la comparación.
- ✓ La **medición de máximo y mínimo teórico** contempla los valores teóricos para efectos de la comparación de la medición con los atributos de calidad especificados. El conocimiento de los valores máximos o mínimos permite el establecimiento claro del grado de cumplimiento de los atributos de calidad.

De forma general el planteamiento anterior presenta los objetivos para efectos de la medición de los atributos de calidad. Sin embargo, en ocasiones, la evaluación de una arquitectura de software no produce valores numéricos que permiten la toma de

decisiones de manera directa. Ante la posibilidad de efectuar evaluaciones a cualquier nivel del proceso de diseño, con distintos niveles de especificación, Kazman propone un esquema general en relación a la evaluación de una arquitectura con respecto a sus atributos de calidad. En este sentido, Kazman y sus colegas afirman que de la evaluación de una Arquitectura no se obtienen respuestas del tipo “si - no”, “bueno – malo” o “6.75 de 10”, sino que explica **cuáles son los puntos de riesgo del diseño evaluado** (2).

Una de las diferencias principales entre los planteamientos de Bosch y Kazman es el enfoque que utilizan para efectos de la evaluación. El método de diseño de arquitecturas planteado por Bosch tiene como principal característica la evaluación explícita de los atributos de calidad de la arquitectura durante el proceso de diseño de la misma. En este sentido, Bosch plantea las técnicas de evaluación: basada en escenarios, basada en simulación, basada en modelos matemáticos y basada en experiencia. Por su parte, Kazman propone que resulta de poco interés la caracterización o medición de atributos de calidad en las fases tempranas del proceso de diseño. Su enfoque se orienta hacia la mitigación de riesgos, ubicando dónde un atributo de calidad de interés se ve afectado por decisiones arquitectónicas (2).

Tanto Bosch como Kazman indican la importancia de la especificación exhaustiva de los atributos de calidad como base para efectos de la evaluación de una arquitectura de software. El punto es definir los atributos de calidad en función de sus metas y su contexto, y no como cantidades absolutas (2).

Características de una Evaluación de Arquitectura

- Es uno de los principales puntos de evaluación dentro del proyecto, ya que errores en ella, pueden traer que el proyecto fracase (5).
- Puede ser realizada por gente Interna o Externa al proyecto, aunque lo más interesante es que sea realizada por gente Externa (Mentores o Arquitectos del Área) (5).

¿Cómo puedo estar seguro que la elección de mi arquitectura es la correcta para mi software?

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).

¿Por qué evaluar una Arquitectura?

“El propósito de realizar evaluaciones a la arquitectura, es para analizar e identificar riesgos potenciales en su estructura y sus propiedades, que puedan afectar al sistema de software resultante, verificar que los requerimientos no funcionales estén presentes en la arquitectura, así como determinar en qué grado se satisfacen los atributos de calidad. Cabe señalar que los requerimientos no funcionales también son llamados atributos de calidad” (4).

Un atributo de calidad es una característica de calidad que afecta a un elemento. Donde el término “característica” se refiere a aspectos no funcionales y el termino “elemento” a componente (4).

- Cuanto más temprano se encuentre un problema en un proyecto de software, mejor (5).
- Realizar una evaluación de la arquitectura es la manera más económica de evitar desastres (5).
- Analizar y evaluar la calidad exigida por los usuarios (5).
- Decisiones de diseño (5)
 - Restricciones de Implementación.
 - Fija la estructura organizacional, tanto del desarrollo, construcción y ejecución del sistema.
 - Logra los atributos de calidad.
 - Permite el prototipado.
 - Permite estimaciones más certeras.
- Abstracción transferible entre sistemas (5)

¿Cuándo una Arquitectura puede ser evaluada?

Es posible realizarla en cualquier momento según Kazman, pero propone dos variantes que agrupan dos etapas distintas: temprano y tarde (2).

- **Temprana.** No es necesario que la arquitectura esté completamente especificada para efectuar la evaluación, y esto abarca desde las fases tempranas de diseño y a lo largo del desarrollo.
- **Tarde.** Cuando ésta se encuentra establecida y la implementación se ha completado. Este es el caso general que se presenta al momento de la adquisición de un sistema ya desarrollado.

Reglas de oro para determinar el momento de realizar evaluaciones (4):

1. Realizar una evaluación cuando el equipo de desarrollo inicia a tomar decisiones que afectan directamente a la arquitectura;...
2. Cuando el costo de no tomar estas decisiones podría tomar más, que el costo de realizar una evaluación.

¿Quiénes participan en una Evaluación?

Generalmente las evaluaciones a la arquitectura se hacen por miembros del equipo de desarrollo, arquitecto, diseñador, entre otros. Sin embargo puede haber también situaciones en las que intervengan personas especialistas en el tema. Otro que también se interesa por los resultados de una evaluación es el cliente, ya que en dependencia de los resultados puede tomar decisiones de continuar o no con el proyecto (4).

Técnicas de Evaluación

Existen un grupo de técnicas para evaluar que se clasifican en cualitativas y cuantitativas (5):

- Técnicas de cuestionamiento o cualitativas. Utilizan preguntas cualitativas para preguntarle a la arquitectura
 - Cuestionarios. Abiertas. Temprana
 - Checklists. Especifico del Dominio de la aplicación.
 - Escenarios. Especificas del Sistema. Arquitectura avanzada.
- Measuring techniques. Sugiere hacerle medidas cuantitativas a la arquitectura.
 - Utiliza métricas arquitectónicas, como acoplamiento, cohesividad en los módulos, profundidad en herencias, modificabilidad.
 - Simulaciones, Prototipos, y Experimentos

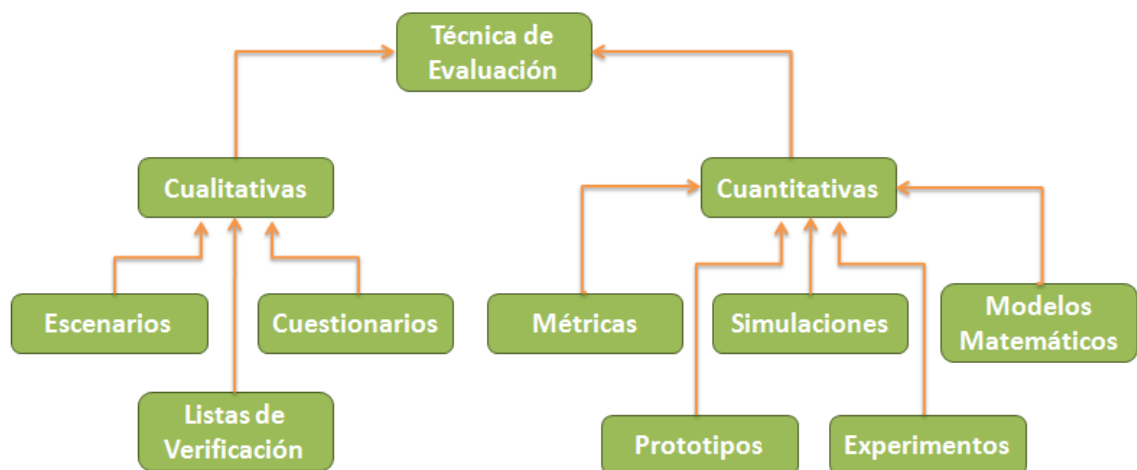


Figura #1: Clasificación de las Técnicas de Evaluación (4)

Por lo regular, las técnicas de evaluación cualitativas son utilizadas cuando la arquitectura está en construcción, mientras que las técnicas de evaluación cuantitativas, se usan cuando la arquitectura ya ha sido implantada (4).

Planear o no una arquitectura

Las evaluaciones pueden ser (4):

- **Planeadas:** Son aquella que han sido planificadas por el ciclo de vida de desarrollo, por lo que es parte de las actividades del proyecto.
- **No planeadas:** Se presenta cuando la arquitectura contiene varios defectos que han sido detectados en las etapas tardías del desarrollo, por ende, realizar una evaluación no planeada, representa retrasos en los tiempos de entrega, así como un incremento en los costos de un proyecto. Por lo que es recomendable que en los proyectos se contemple realizar una o varias evaluaciones a la arquitectura.

¿Por qué cualidades puede ser evaluada una Arquitectura?

A grandes rasgos, Bass establece una clasificación de los atributos de calidad en dos categorías (2):

- **Observables vía ejecución:** aquellos atributos que se determinan del comportamiento del sistema en tiempo de ejecución. La descripción de algunos de estos atributos se presenta en la tabla 1.
- **No observables vía ejecución:** aquellos atributos que se establecen durante el desarrollo del sistema. La descripción de algunos de estos atributos se presenta en la tabla 2.

Tabla #1. Descripción de atributos de calidad observables vía ejecución (2)

Atributo de Calidad	Descripción
Disponibilidad	Es la medida de disponibilidad del sistema para el uso (Barbacci et al, 1995).
Confidencialidad	Es la ausencia de acceso no autorizado a la información (Barbacci et al, 1995).
Funcionalidad	Habilidad del sistema para realizar el trabajo para el cual fue concebido (Kazman et al., 2001).
Desempeño	Es el grado en el cual un sistema o componente cumple con sus funciones designadas, dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria. (IEEE 610.12).
Confiabilidad	Es la medida de la habilidad de un sistema a mantenerse operativo a lo largo

	del tiempo (Barbacci et al., 1995).
Seguridad externa	Ausencia de consecuencias catastróficas en el ambiente. Es la medida de ausencia de errores que generan pérdidas de información (Barbacci et al, 1995).
Seguridad interna	Es la medida de la habilidad del sistema para resistir a intentos de uso no autorizados y negación del servicio, mientras se sirve a usuarios legítimos (Kazman et al., 2001).

Tabla #2. Descripción de atributos de calidad no observables vía ejecución (2)

Atributo de Calidad	Descripción
Configurabilidad	Posibilidad que se otorga a un usuario experto a realizar ciertos cambios al sistema (Bosch et al., 1999).
Integrabilidad	Es la medida en que trabajan correctamente componentes del sistema que fueron desarrollados separadamente al ser integrados. (Bass et al. 1998)
Integridad	Es la ausencia de alteraciones inapropiadas de la información (Barbacci et al., 1995).
Interoperabilidad	Es la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema. Es un tipo especial de <i>integrabilidad</i> (Bass et al. 1998)
Modificabilidad	Es la habilidad de realizar cambios futuros al sistema. (Bosch et al. 1999).
Mantenibilidad	Es la capacidad de someter a un sistema a reparaciones y evolución (Barbacci et al., 1995). Capacidad de modificar el sistema de manera rápida y a bajo costo (Bosch et al. 1999).
Portabilidad	Es la habilidad del sistema para ser ejecutado en diferentes ambientes de computación. Estos ambientes pueden ser hardware, software o una combinación de los dos (Kazman et al., 2001).
Reusabilidad	Es la capacidad de diseñar un sistema de forma tal que su estructura o parte de sus componentes puedan ser reutilizados en futuras aplicaciones (Bass et al. 1998).
Escalabilidad	Es el grado con el que se pueden ampliar el diseño arquitectónico, de datos o procedimental (Pressman, 2002).
Capacidad de Prueba	Es la medida de la facilidad con la que el software, al ser sometido a una serie de pruebas, puede demostrar sus fallas. Es la probabilidad de que, asumiendo que tiene al menos una falla, el software fallará en su próxima ejecución de prueba (Bass et al. 1998).

¿Cuáles son los beneficios de realizar una evaluación arquitectónica? (5)

- Financieros.
- Reúne a los stakeholders.
- Fuerza una articulación en las metas específicas de calidad.
- Fuerza una mejora a la documentación de la arquitectura.
- Mejora la arquitectura.
- Detección temprana de problemas.
- Valido requerimiento y los priorizo.
- Recolecta los fundamentos y las decisiones arquitectónicas tomadas.

¿Qué resultados produce la evaluación de una Arquitectura?

Una vez que se ha efectuada la evaluación se debe elaborar un reporte. Que debe presentarse como un documento preliminar, con la finalidad de que se corrija por las personas que participaron en la evaluación. El contenido del reporte responde a dos tipos de preguntas (4):

- ¿Se ha diseñado la arquitectura más apropiada para el sistema?
- ¿Cuál de las arquitecturas propuestas es la más apropiada para el sistema a construir?

Además de responder estas preguntas, el reporte también indica el grado en que se cumplieron los atributos de calidad.

¿Cuáles son las salidas de una evaluación arquitectónica? (6)

- Lista priorizada de los atributos de calidad requeridos para la arquitectura que está siendo evaluada.
- Riesgos y no riesgos.

Pasos Básicos para una Evaluación (5)

- **Preparación.**
 - Contexto.
 - Evaluador.
 - Alcance.
 - Representación de la Arquitectura.
- **Realización.**
 - Presentar el problema de negocio a resolver y la arquitectura planteada.
 - Evalúa, el costo, la funcionalidad, los atributos de calidad.
 - Revisan requerimientos y posibles cambios.

- Discuten problemas y observaciones.
- **Generar y distribuir los resultados**
 - Issues y Recomendaciones.
 - Análisis de riesgo.

Métodos de Evaluación de Arquitecturas

- **ATAM** (*Architecture Trade-off Analysis Method*): está inspirado en tres áreas distintas: los estilos arquitectónicos, el análisis de atributos de calidad y el método de evaluación SAAM (*Software Architecture Analysis Method*). El nombre del método ATAM surge del hecho de que revela la forma en que una arquitectura específica satisface ciertos atributos de calidad, y provee una visión de cómo los atributos de calidad interactúan con otros.

El método se concentra en la identificación de los estilos arquitectónicos o enfoques arquitectónicos utilizados. Estos elementos representan los medios empleados por la arquitectura para alcanzar los atributos de calidad, así como también permiten describir la forma en la que el sistema puede crecer, responder a cambios, e integrarse con otros sistemas, entre otros (2).

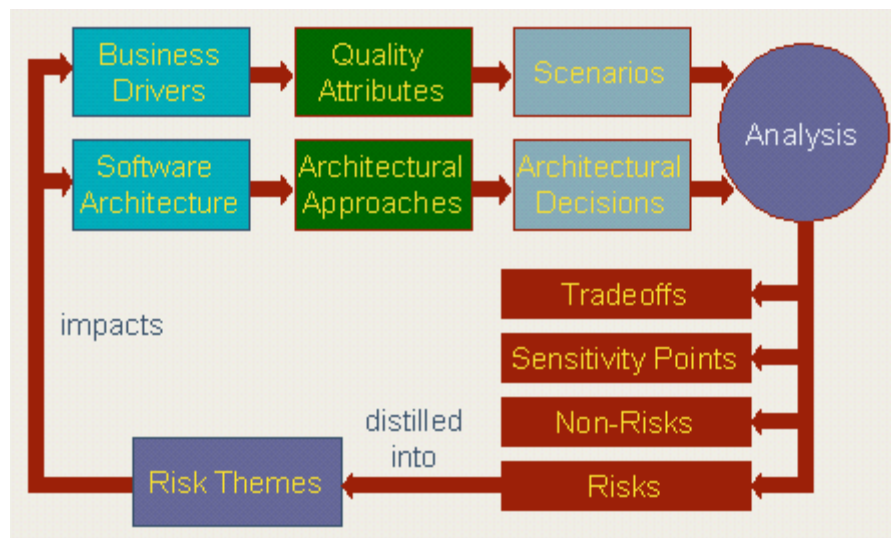


Figura #2. Flujo conceptual del método ATAM (7)

El método de evaluación ATAM comprende nueve pasos, agrupados en cuatro fases (Presentación, Investigación y Análisis, Pruebas y Reporte).

- **Bosch (2000).** Que plantea que: “El proceso de evaluación debe ser visto como una actividad iterativa, que forma parte del proceso de diseño, también iterativo. Una vez que la arquitectura es evaluada, pasa a una fase de transformación, asumiendo que no satisface todos los requerimientos. Luego, la arquitectura transformada es evaluada de nuevo” (2). Este método consta de 5 pasos divididos en dos etapas.
- **ADR (Active Design Review).** “ADR es utilizado para la evaluación de diseños detallados de unidades del software como los componentes o módulos. Las preguntas giran en torno a la calidad y completitud de la documentación y la suficiencia, el ajuste y la conveniencia de los servicios que provee el diseño propuesto” (2).
- **ARID (Active Reviews for Intermediate Design).** ARID es un método de bajo costo y gran beneficio, el mismo es conveniente para realizar la evaluación de diseños parciales en las etapas tempranas del desarrollo (2). ARID es un híbrido entre ADR y ATAM (8). Se basa en ensamblar el diseño de los stakeholders para articular los escenarios de usos importantes o significativos, y probar el diseño para ver si satisface los escenarios. Como resultado de la aplicación de dicho procedimiento se obtiene un diseño de alta fidelidad acompañado de una alta familiarización con el diseño de los stakeholders. Este método consta de 9 pasos agrupados en dos fases (Actividades Previas y Evaluación) (9).
- **Losavio (2003):** Es un método para evaluar y comparar arquitecturas de software candidatas, que hace uso del modelo de especificación de atributos e calidad adaptado del modelo ISO/IEC 9126. La especificación de los atributos de calidad haciendo uso de un modelo basado en estándares internacionales ofrece una vista amplia y global de los atributos de calidad, tanto a usuarios como arquitectos del sistema, para efectos de la evaluación (2). El método contempla siete actividades.

Tabla #3. Comparación entre Métodos de Evaluación (2).

	ATAM	SAAM	ARID	Bosch(2000)	Losavio(2003)
Atributos de Calidad	Modificabilidad	Modificabilidad	Conveniencia del diseño	Seleccionados por el arquitecto, de acuerdo a la importancia sobre el sistema	Funcionabilidad
Contemplados	Seguridad Confiabilidad Desempeño	Funcionabilidad	evaluado		Confiabilidad Usabilidad Eficiencia Mantenimiento

						Portabilidad
Objetos Analizados	Estilos Arquitectónicos, Documentación, Flujo de Datos y Vistas Arquitectónicas	Documentación, y Vistas Arquitectónicas	Especificación de los componentes	Estilos Arquitectónicos, Vistas Arquitectónicas, Patrones Arquitectónicos, Patrones de Diseño y Patrones de Idioma	Especificación de Atributos de Calidad	
Etapas del Proyecto que se Aplica	Luego que el diseño de la arquitectura ha sido establecido	Luego que la arquitectura cuenta con funcionalidad ubicada en módulos	A lo largo del diseño de la arquitectura	Luego que el diseño de la arquitectura ha sido establecido	Luego que el diseño de la arquitectura ha sido establecido	
Enfoques Utilizados	Árbol de Utilidad y lluvia de ideas para articular los requerimientos de calidad. Análisis arquitectónico que detecta puntos	Lluvia de ideas para escenarios y articular los requerimientos de calidad. Análisis de los escenarios para verificar funcionalidad o estimar el costo de los	Revisiones de diseño, lluvia de ideas para obtener escenarios.	Análisis de perfiles (perfiles)	Análisis y comparación con de los resultados para las arquitecturas candidatas	

sensibles, cambios.
puntos de
balance y
riesgos.

Existen otros métodos de evaluación de arquitectura que evalúan un atributo de calidad específico:

- **ALMA** (*Architecture Level Modifiability Analysis*). Este método es el resultado de los trabajos de investigación de *Brengtsson* y *Lassing*. El atributo de calidad que analiza este método es la facilidad de modificación. Esto se refiere a la capacidad de un sistema para ser ajustado debido a cambios en los requerimientos, o en el entorno, así como la adición de nuevas funcionalidades (9).

ALMA es un método de evaluación orientado a metas, que se apoya en el uso de escenarios de cambio, los cuales describen los eventos posibles que provocarían cambios al sistema, y como se llevarían a cabo estos. El mismo consta de cinco pasos como se muestra en la siguiente figura (9).

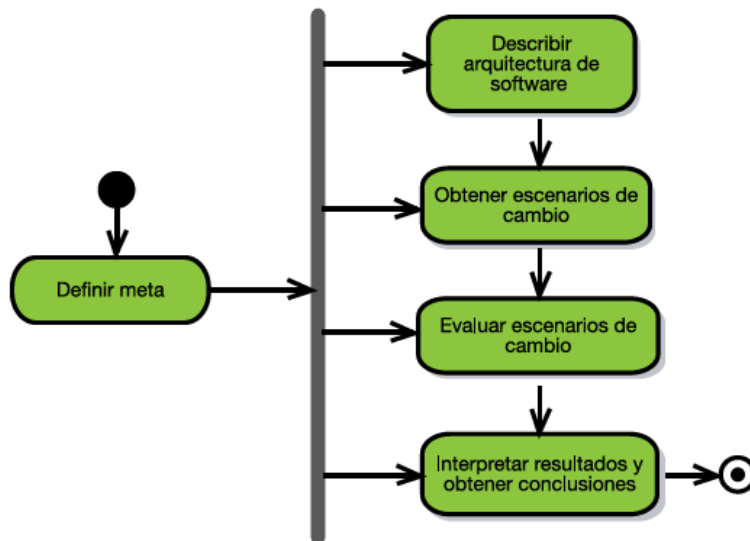


Figura #3. Método ALMA (9)

- **PASA** (*Performance Assessment of Software Architecture*). El atributo de calidad que analiza este método es el desempeño. Que se interesa en saber qué tanto tiempo le toma al

sistema software responder cuando una o varios eventos ocurren, así como determinar el número de eventos procesados en un intervalo de tiempo dado. PASA es el resultado del trabajo de Williams y Smith, el mismo utiliza diversas técnicas de evaluación, tales como la aplicación de estilos arquitectónicos, anti-patrones, guías de diseño y modelos (9).

Este método también se basa en escenarios y puede aplicarse de forma temprana o tardía. Los escenarios generados en este método sirven como punto de partida para la construcción de modelos de desempeño (9).

Uno de los requisitos o precondiciones que presenta, es que la arquitectura debe estar previamente documentada y en caso de que no esté completa se debe extraer el resto de la información a los miembros del equipo (9).

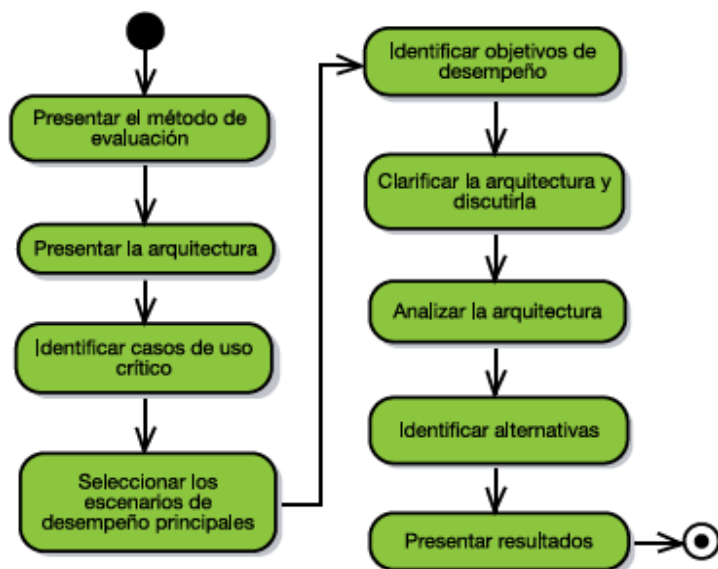


Figura #4. Método PASA (9)

Las personas involucradas durante el proceso de evaluación son: el arquitecto, equipo de desarrollo y en algunos momentos el gerente(s) del proyecto. La evaluación empleando este procedimiento puede durar una semana si se traba de forma intensivamente. Es considerado un método de evaluación maduro ya que ha sido probado en varios dominios de aplicación como sistemas web, aplicaciones financieros y sistemas en tiempo real (9).

- **SALUTA** (Scenario based Architecture Level Usability Analysis). Es el primer método desarrollado para evaluar arquitecturas desde la perspectiva de la facilidad del uso del sistema, siendo el resultado de los estudios de Folmer y Gulp (9).

Este método hace uso de marcos de referencia que expresan las relaciones que existen entre facilidad de uso y Arquitectura de Software. Dichos marcos de referencias incluyen un conjunto integrado de soluciones de diseño como patrones de diseños u propiedades que tienen un efecto positivo sobre la facilidad de uso en un sistema software (9).

SALUTA analiza cuatro atributos que están directamente relacionados con la facilidad de uso de un sistema de software: facilidad de aprendizaje, eficiencia de uso, confiabilidad y satisfacción. El mismo se basa al igual que los dos métodos analizados anteriormente en escenarios, que en este caso, son escenarios de uso que agrupan uno a más perfiles de uso valga la redundancia, donde cada uno representa la facilidad de uso requerida por el sistema (9).

Se recomienda utilizarlo una vez que se ha especificado la arquitectura, pero antes de implementar (9). La misma cuenta consta de cuatro pasos como se muestra a continuación:

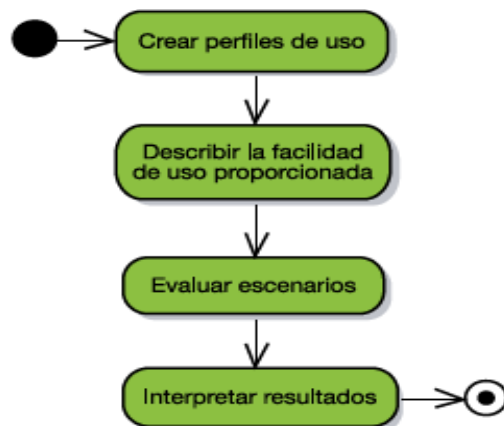


Figura #5. Método SALUTA (9).

Las personas involucradas durante el proceso de evaluación son: el arquitecto de software, ingenieros de requerimientos o ingenieros responsable por la facilidad de uso (9).

- **SNA** (Survivable Network Analysis). Es un método desarrollado por el CERT (*Computer Emergency Response Team*) que forma parte del SEI (*Software Engineering Institute*). Este método ayuda a identificar la capacidad de supervivencia en un sistema, analizando

su arquitectura. La supervivencia es la capacidad que tiene un sistema para completar su misión a tiempo, ante la presencia de ataques, fallas o accidentes. Para evaluar esta supervivencia SNA utiliza tres propiedades claves: Resistencia, Reconocimiento y Recuperación (9).

Este procedimiento puede ser realizado después de la especificación de la arquitectura, durante la implementación de esta, o posteriormente. SNA se basa en la técnica de escenarios de uso, e identifica dos tipos de escenarios (9):

1. Escenarios normales, que se componen de una serie de pasos donde los usuarios invocan servicios y obtienen acceso a activos, tales como bases de datos.
2. Escenarios de intrusión, en los que se representan diferentes tipos de ataques al sistema.

Las personas involucradas durante la realización del método son: el arquitecto de software, el diseñador principal, los propietarios del sistema y usuarios del mismo. Como resultado de esta evaluación se obtienen: un documento que recoge todas las modificaciones recomendaciones a la arquitectura, acompañadas del mapa de supervivencia (9).

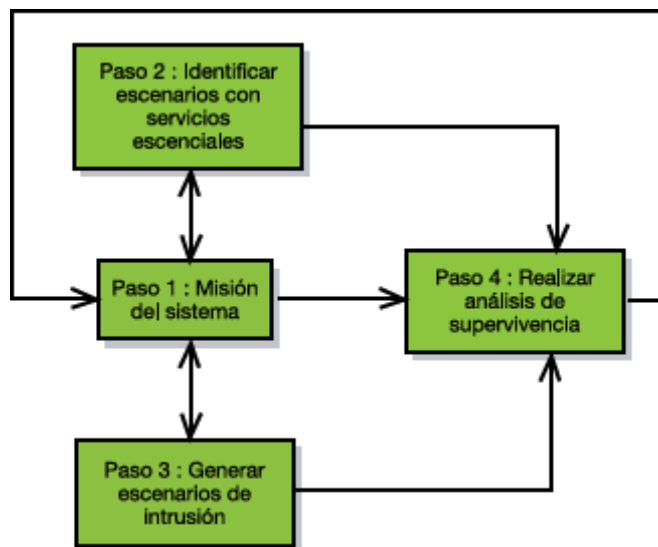


Figura #6. Método SNA (9).

La siguiente tabla presenta a manera de resumen, un cuadro comparativo entre estos métodos analizados anteriormente:

Tabla #4. Comparación entre los métodos ALMA, PASA, SALUTA y SNA (9).

	ALMA	PASA	SALUTA	SNA
Meta	Predecir el costo de mantenimiento, evaluar riesgos, comparación entre arquitecturas.	Analizar la arquitectura con respecto a los objetivos de desempeño de un sistema.	Predecir la facilidad de uso en un sistema analizando la arquitectura.	Identificar la capacidad de supervivencia en un sistema ante presencia de ataques, fallas o accidentes.
Atributo de calidad	Facilidad de modificación	Desempeño	Facilidad de uso	Supervivencia
Técnica de evaluación	Escenarios de cambio	Escenarios	Escenarios de uso	Escenarios de uso normales, escenarios de intrusión.
Entradas	Especificación de la arquitectura, requerimientos no funcionales.	Especificación de la arquitectura.	Especificación de la arquitectura, requerimientos no funcionales relacionados con la facilidad de uso.	Especificación de la arquitectura
Salidas	Dependiendo de la meta de evaluación se generan los resultados.	Hallazgos encontrados, pasos específicos a seguir y recomendaciones.	Grado de facilidad de uso que soporta la arquitectura evaluada.	Modificaciones recomendadas a arquitectura y mapa de supervivencia.
Personas involucradas	Arquitecto y equipo de desarrollo	Arquitecto, equipo de desarrollo y administradores del proyecto.	Arquitecto, ingenieros de requerimientos o ingenieros responsables por la facilidad de uso.	Arquitecto, diseñador principal, propietarios del sistema, usuarios
Duración	No especificado	7 días	No especificado	No especificado
Validación del método	Sistemas de control embebido, sistemas médicos, telecomunicaciones, sistemas administrativos.	Sistemas basados en Web, aplicaciones financieras y sistemas en tiempo real.	Algunos casos de estudio que incluyen principalmente sistemas Web.	Sistemas comerciales y de gobierno

CONCLUSIONES PARCIALES

- El método ATAM evalúa con más profundidad, en relación con otros métodos, cuestiones referentes a la arquitectura, como son: los atributos de calidad.
- Hasta el momento se han presentado varios métodos de evaluación de arquitectura algunos más completos que otros, pero independientemente de esto todos tienen algo en común y es que utilizan la técnica de escenarios como vía de constatar en qué medida la arquitectura responde a los atributos de calidad requeridos por el sistema.

CAPÍTULO 2. Resultados y Discusión

Introducción

En este capítulo se propone un procedimiento para evaluar Arquitecturas de Software Basadas en Componentes (ASBC) con el objetivo de que sea aplicado en Arquitecturas Orientadas a Servicios (SOA). Este análisis se realiza desde el punto de vista, que bajo

ciertas y determinadas situaciones o condiciones, se puede ver un servicio como un componente.

Objetivos

El objetivo que se persigue en este capítulo es describir y analizar paso a paso como funciona el método propuesto, quiénes participan, y los instrumentos y técnicas de evaluación empleadas para constatar el grado de cumplimiento con los atributos de calidad del sistema.

Propuesta de Procedimiento

Como resultado del análisis de la comparación de diferentes métodos, se observó que el Architecture Tradeoffs Analysis Method (ATAM), tiene un conjunto de pasos, un equipo de evaluación y un conjunto de salidas mejor definidas y no presenta ninguna restricción con respecto a la característica de calidad a evaluar. El MECABIC está inspirado en ATAM, aunque con el objetivo de facilitar su aplicación sobre ASBC se incluyó en algunos de sus pasos un enfoque de integración de elementos de diseño, inspirado en la construcción de partes arquitectónicas adoptado por el Architecture Based Design (ABD). Además se propone un árbol de utilidad inicial basado en el modelo de calidad ISO-9126 instanciado para Arquitectura de Software propuesto por Losavio. La adopción de este modelo por parte del MECABIC permite concentrarse en características que dependen exclusivamente de la arquitectura, y al ser un estándar facilita la correspondencia con características de calidad consideradas por los métodos estudiados. Los escenarios incluidos en este árbol son específicos para aplicaciones basadas en componentes (1).

Equipo de Evaluación

En el método MECABIC participan tres grupos de trabajo

Tabla #5. Grupos participantes en el método MECABIC (1).

Equipo	Definición	Fases en las que participan
Arquitectos	Responsables de generar y documentar una Arquitectura de Software para el sistema estudiado.	Todas
Evaluador	Integrado por personas expertas en asuntos de calidad quienes guiarán el proceso de evaluación de la arquitectura	Todas

Instrumentos y técnicas de evaluación

Para la especificación de la calidad se hace uso de un árbol de utilidad, el cual tiene como nodo raíz la “bondad” o “utilidad” del sistema. En el segundo nivel del árbol se encuentran los atributos de calidad. Las hojas del árbol de utilidad son escenarios, los cuales representan mecanismos mediante los cuales extensas (y ambiguas) declaraciones de cualidades son hechas específicas y posibles de evaluar. La generación del árbol de calidad incluye un paso que permite establecer prioridades. Para el análisis de la arquitectura se utiliza la técnica de escenarios, soportada por cuestionarios, para identificar lo que desean los participantes (1).

Fases del Método

1. En la primera fase, de *presentación*, se da a conocer el método entre todos los grupos, el sistema y su organización, y finalmente se presenta cuál es la arquitectura que debe ser evaluada (1).
2. La segunda es de *investigación y análisis*, y en ella se determina de qué manera se va estudiar la arquitectura, se pide a los *stakeholders* que expresen de una manera precisa que escenarios de calidad se desean y se analiza si la arquitectura es apropiada o se requieren modificaciones para que lo sea. En esta fase sólo participan el grupo evaluador y grupo de arquitectos (1).
3. La tercera fase es *de prueba*, consiste en la revisión de la segunda fase y en ella participan todos los grupos (1).
4. En la última fase se lleva a cabo a presentación de los resultados. En esta fase participan todos los grupos (1).

A continuación se explican cada una de las fases con sus respectivos pasos.

1. Fase de presentación (1)

En esta fase se realiza una presentación del método MECABIC para que los participantes comprendan en qué punto de la aplicación del método se encuentran en cada momento. También se da a conocer la arquitectura inicial a evaluar y qué características de calidad se esperan lograr.

1. Presentación del método.

En el primer paso el director de evaluación presenta el método ante los participantes con el proyecto. Durante una reunión se explica el proceso que debe cumplir cada participante del proyecto, se responden preguntas y se establecen las expectativas y el contexto sobre las actividades o tareas restantes. Este paso tiene como finalidad que todos los *stakeholders* comprendan la secuencia de los pasos a seguir, los instrumentos utilizados en cada paso y las salidas del método. Se pide a los stakeholders que comenten sus expectativas o que hagan preguntas particulares acerca de lo que esperan de la aplicación del método.

2. Presentación de las directrices del negocio.

Cada persona que posee una responsabilidad sobre la evaluación (representantes del proyecto, miembros del equipo, etc.), necesitan comprender el contexto del sistema y los aspectos primarios del contexto del negocio que motivan su desarrollo. En este paso se explica a los stakeholders el contexto del sistema y los requerimientos del negocio dentro del cual va a funcionar el sistema. Algunos tópicos que debería contener esta presentación son: las funcionalidades más importantes del sistema, los objetivos del negocio y el contexto relacionado con el proyecto, el grupo dirigente de los stakeholders, las directrices arquitectónicas (Requerimientos de calidad a ser satisfechos por la arquitectura), restricciones técnicas, gerenciales, económicas y políticas y obertura de la evaluación y límites del sistema.

3. Presentación de la arquitectura.

En este paso, el arquitecto o grupo de representantes, debe explicar a los stakeholders cuál es la arquitectura evaluada. Debe contener una clara diferenciación estructural, la cual deberá mostrar claramente las relaciones entre componentes de la arquitectura y las diferentes granularidades involucradas. En dicha presentación, el arquitecto cubre restricciones técnicas (Sistema operativo, hardware, otros sistemas con el cual el sistema debe interactuar, etc.) y los alcances arquitectónicos usados para alcanzar los

requerimientos. El arquitecto debe transmitir lo esencial y de esta manera abrevia la información de la arquitectura que requiere el equipo.

2. Fase de Investigación y análisis (1)

En esta fase se identifican elementos de diseño que ayuden a analizar la arquitectura, se especifican los requerimientos planteados durante el paso 2 y se analiza cómo responden los elementos de diseño considerados a los requerimientos.

4. Identificación de elementos de diseño.

En este paso se contemplan alternativas de diseño de la arquitectura, que posteriormente serán analizadas para ver cómo responden a los requerimientos de calidad elicitados. La arquitectura debe ser evaluada completamente desde el sistema con sus componentes de mayor granularidad, hasta el mínimo nivel de granularidad que corresponde a los componentes. En este método se propone identificar elementos de diseño dentro de una arquitectura basada en componentes como un componente y el conjunto de componentes con los cuales interactúa, un conjunto de asociaciones que sea de interés sobre alguna característica de calidad particular o conjunto de decisiones que tenga influencia considerable sobre otros elementos de diseño. Para identificar un elemento de diseño es necesario considerar los servicios ofrecidos por los componentes disponibles y las diferentes opciones para definir los servicios de los componentes desarrollados para el sistema. La manera de documentar los elementos de diseño depende de la importancia que pueda tener dentro de la evaluación, del conjunto de decisiones o adaptaciones que sea necesario realizar, y del conocimiento de los stakeholders presentes en la evaluación.

5. Generación del árbol de utilidad.

El MECABIC proporciona un árbol de utilidad inicial específico para ASBC a partir del cual seleccionan un conjunto de escenarios de interés (ver la tabla#6), el cual está basado en el modelo de calidad ISO 9126-1 para arquitecturas de software propuesto por Losavio. Se asume que al aplicar el

método, las funcionalidades presentes en el sistema son las que necesitan los usuarios, y en consecuencia, no se incluye escenarios de aptitud (subcaracterística de funcionalidad). Posteriormente, se consideran escenarios no contemplados en el árbol inicial de utilidad.

Tabla #6. Subconjunto del árbol de utilidad inicial propuesto por el método (1)

Característica	Sub-característica	Escenario
Funcionalidad	Interoperabilidad	El sistema posee componentes capaces de leer datos provenientes de otros sistemas.
		El sistema posee componentes capaces de producir datos para otro sistema.
	Precisión	Los resultados ofrecidos por los componentes sistema son exactos. La comunicación entre los componentes no altera la exactitud de los datos
	Seguridad	El sistema detecta la actuación de un intruso e impide acceso a los componentes que manejen información sensible El sistema asegura que los componentes no pierdan datos ante un ataque (interno o externo).
Fiabilidad	Obediencia (Compliance)	Los componentes respetan un estándar de fiabilidad. La comunicación entre los componentes no viola los estándares de fiabilidad.
	Madurez	Los componentes del sistema manejan entradas de datos de datos incorrectas.
	Tolerancia a fallas	Todas las operaciones ejecutadas por los componentes se realizan correctamente bajo condiciones adversas.
Eficiencia	Capacidad de restablecimiento o recuperación	Los componentes del sistema no fallan bajo ciertas condiciones especificadas. Ante problemas con el ambiente un subconjunto determinado de los componentes puede continuar prestando sus servicios.
	Tiempo de comportamiento Recursos utilizados	El sistema debe recibir los servicios de sus componentes en el transcurso de un tiempo indicado. Los componentes pueden compartir recursos adecuadamente. El sistema controla que ningún componente se quede sin recursos cuando los necesita.
Mantenibilidad	Habilidad de cambio, estabilidad, prueba	Es posible verificar el estado de los componentes del sistema. El sistema brinda facilidad para adaptar un componente.

		El sistema debe facilitar la sustitución/adaptación de un componente.
Portabilidad	Adaptabilidad	El sistema debe continuar funcionando correctamente aun cuando los servicios de los componentes provistos por el ambiente varíen.
	Capacidad de Instalación	Los componentes pueden instalarse fácilmente en todos los ambientes donde debe funcionar.
	Co-existencia	Los componentes manejan adecuadamente recursos compartidos del sistema.

A continuación se describen los pasos a seguir para obtener el árbol de utilidad.

5.1. Seleccionar los escenarios iniciales a considerar.

En este paso el grupo evaluador presenta los diferentes escenarios a considerar al resto de los participantes y se decide cuáles son los que serán incluidos en el árbol de utilidad.

5.2. Proposición de escenarios no contemplados.

Este paso busca brindar a los stakeholders la posibilidad de que verifiquen si es necesario incluir en el árbol de utilidad algún otro escenario. Si se determinan escenarios de calidad que cuenten con un alto consenso de los participantes deberían ser incluidos directamente dentro del árbol de utilidad. La selección del resto de los escenarios puede realizarse por votación como propone originalmente el ATAM. Luego se organizan los escenarios de calidad introducidos no contemplados en la tabla de generación del árbol de utilidad inicial por características de calidad. Debido a que se evalúa un conjunto de características de interés no deberían estar presentes escenarios de calidad para todas las características de calidad conocidas. Una vez obtenido los escenarios a incluir árbol de utilidad se procede a priorizarlos.

5.3. Priorización de los escenarios de calidad.

En este paso el objetivo se debe ordenar los escenarios ya que de esta manera los arquitectos cuentan con más orientación para tomar decisiones

arquitectónicas, y los stakeholders pueden estar más conscientes de lo que esperan del sistema, y obtener una idea sobre en qué medida va a ser satisfecho. Una vez que se ha decidido incluir en el árbol de utilidad los escenarios más importantes, se procede a asignar un orden a los escenarios de calidad de un sistema utilizando dos criterios, a saber:

- a) Evaluar el riesgo de no considerar el escenario. Se deben responder las preguntas: ¿qué pasa si este escenario no se cumple?, ¿cuántas personas se ven afectadas?, ¿es posible compensar el no responder a este escenario?
- b) Evaluar el esfuerzo necesario para lograr cumplir con el escenario. En este caso se determina que cambios o integraciones de nuevos componentes se deben realizar para satisfacer el escenario. Los escenarios más difíciles son aquellos en los que se debe consumir más tiempo de análisis y el resto debe ser guardado como parte del registro.

Finalmente, se construye el árbol de utilidad con las salidas del paso anterior. La prioridad del escenario define en este método como un par (X, Y) en el cual X define el esfuerzo de satisfacer el escenario, y la Y indica los riesgos que se corren al excluirlos del árbol de utilidad. Los posibles valores para X e Y son A (Alto), B (Bajo) y M (Medio). El árbol de utilidad generado se toma como un plan para el resto de la evaluación que realiza el método. Indica además al equipo evaluador dónde ocupar su tiempo y dónde probar aproximaciones y riesgos arquitectónicos.

6. Análisis de elementos de diseño para ASBC.

En este paso se analizan los elementos de diseño identificados en el paso 4 o de nuevos elementos de diseño que puedan ser identificados, para determinar cómo influyen sobre la realización de los escenarios de calidad presentes en el árbol de utilidad generado en el paso 5.

6.1. Análisis de elementos de diseño estudiados en el paso 4.

Dependiendo del tipo de elemento de diseño, la evaluación será diferente. Los componentes proveen un conjunto de puertos que se deben ir extendiendo para proporcionar nuevos servicios, los cuales a su vez

pueden ser ofrecidos para que otros componentes los extiendan. Se debe evaluar un conjunto de opciones que indiquen una manera de definir una relación entre puertos (planteadas en el paso 4 y que pueden ser extendidas en este paso, considerando el árbol de utilidad generado), de acuerdo a algún criterio de calidad. Los escenarios de calidad deben ser aplicados a la arquitectura que ha sido generada. El equipo de evaluación se orienta con las preguntas de análisis propuestas por el método para determinar decisiones sobre la arquitectura. Se debe preguntar si las decisiones tomadas permiten alcanzar los escenarios de calidad planteados. Si la respuesta es negativa se deben reconsiderar cualquiera de las decisiones que han sido hechas. Algunas de las preguntas propuestas por el método para analizar los elementos de diseño se muestran a continuación.

Tabla #7. Algunas preguntas para analizar los elementos de diseño identificados (1)

Característica	Sub-característica	Preguntas de análisis
Funcionalidad	Precisión	¿Puede la comunicación entre los componentes introducir imprecisiones en los servicios ofrecidos por los componentes?
	Interoperabilidad	¿Dónde se encuentran los componentes que permiten al sistema interactuar con otros sistemas?
Fiabilidad	Madurez	¿Existen decisiones para minimizar el manejo incorrecto de datos en la comunicación entre componentes?
	Tolerancia a fallas	¿Cómo se detecta el funcionamiento incorrecto de un componente?
Eficiencia	Tiempo de comportamiento	¿Cómo es la relación entre el número de componentes de las diferentes partes de la arquitectura?
Mantenibilidad	Habilidad de cambio, estabilidad, prueba	¿Cómo se verifica el funcionamiento correcto de un componente?
		¿Cómo se verifica el estado de una comunicación entre componentes?
		¿Cómo se facilita el reemplazo de un componente?
Portabilidad	Capacidad de Instalación	¿Existe un mecanismo para determinar si todos los componentes del sistema se encuentran correctamente instalados?
	Co-existencia	¿Existe alguna manera de identificar las reglas para interactuar con componentes de sistemas externos a la arquitectura?

6.2.Documentación de los puntos de negociación, puntos de equilibrio, riesgos, no-riesgos y asuntos no resueltos.

Las decisiones identificadas en el paso 6.1, pueden relacionarse con determinados elementos de diseño. Se deben estudiar los riesgos que introduce la decisión en particular, o si ésta contribuye a algún riesgo ya determinado. También se pueden documentar decisiones que no han sido tomadas o asuntos no resueltos que a juicio del equipo de evaluación pueden ser resueltos en un futuro estudiando con más detalle el elemento de diseño considerado.

3. Fase de prueba (1)

Como resultado de la aplicación de las fases 1 y 2 se tienen un conjunto de decisiones arquitectónicas documentadas que fueron realizadas considerando el árbol de utilidad generado. En esta fase se confrontan las decisiones producidas hasta el momento, y se trabaja con todos los involucrados del sistema para producir la arquitectura final.

7. Revisión del árbol de utilidad.

El MECABIC propone utilizar escenarios de calidad para representar los intereses de todos los grupos de la evaluación y confirmar que se han evaluado los requerimientos deseados de calidad. El equipo de evaluación puede facilitar la elaboración de escenarios haciendo uso del conjunto de pasos propuestos en el paso 6. Los escenarios del árbol de utilidad que no hayan sido considerados, son colocados por los stakeholders en el paquete de tormenta de ideas, lo que les da la oportunidad de revisar escenarios poco atendidos. Los escenarios generados se comparan con la lista inicialmente considerada en el árbol de utilidad. En caso de que la consideración y priorización coincida, entonces se puede decir los escenarios evaluados son adecuados para el grupo de interesados en el sistema. Si no coinciden, los nuevos escenarios y/o su prioridad deben ser reconsiderados. Cuando no se logra un acuerdo entre los dos grupos de stakeholders posiblemente no se representaron adecuadamente los intereses de los involucrados. Los stakeholders deben analizar también los escenarios que ya han sido evaluados, para verificar que hayan recibido la

importancia adecuada. Una vez que el nuevo escenario ha sido considerado se pueden presentar tres casos:

- a) El escenario duplica un escenario ya considerado en el árbol de utilidad.
- b) El escenario pasa a ser una hoja de una rama ya existente.
- c) No existe rama para el escenario debido a que no había sido considerado previamente.

Los primeros dos casos sugieren que la arquitectura de software ha sido creada de acuerdo con las expectativas de los stakeholders, mientras que en el tercer caso, se ha fallado al considerar algún aspecto de calidad importante y puede existir un riesgo a documentar.

8. Revisión de los elementos de diseño definidos.

El equipo evaluador debe estudiar de qué manera los elementos de diseño analizados en el paso 6, promueven los escenarios propuestos por el nuevo grupo de stakeholders. En caso que no promuevan las características de calidad deseadas, deben ser reconsiderados.

4. Fase de Generación de la arquitectura final y reporte (1)

En este momento se cuenta ya con un análisis de todas las decisiones que se han producido hasta el momento. Si no hay conflictos y se puede concluir que existe una arquitectura adecuada para los requerimientos de calidad de los stakeholders involucrados en la evaluación, entonces se puede pasar al paso 10 directamente. Si por el contrario en algún elemento de diseño existen decisiones en las que resulte favorecido algún requerimiento, entonces los stakeholders son los que tienen que determinar o acordar cuáles requerimientos favorecer.

9. Generación de los acuerdos.

En este paso se decide cuál será la arquitectura adoptada por el sistema. Para ello se debe buscar un consenso en cuanto a las opciones que existan, en caso de que no se haya identificado alguna notablemente mejor. Si existen requerimientos de calidad que no han sido logrados se debe brindar la posibilidad de que los stakeholders acepten replantear los requerimientos de calidad que no hayan podido ser alcanzados, para aprovechar posibles ventajas

de la arquitectura. Para este momento los stakeholders tienen una idea de cuáles beneficios pueden ser obtenidos mediante las alternativas que se han estudiado. Esta es una negociación crítica, ya que de fallar, implicaría la necesidad de replantear la arquitectura, y de tener éxito hay que cuidar que los requerimientos de calidad no resueltos sean equivalentes, para que los stakeholders estén contentos con el sistema. Finalmente, se documentan todos aquellos requerimientos de calidad para los cuales no es posible encontrar una solución, justificando las razones.

10. Presentación de los resultados.

Para finalizar la aplicación del método se presenta un resumen de la aplicación del método, de forma oral y/o escrita. Se deben incluir entonces, los siguientes documentos a partir de los cuales se inició la evaluación:

- a) Contexto del negocio.
- b) Requerimientos de Calidad.
- c) Restricciones.
- d) Arquitectura producida.
- e) Análisis de elementos de diseño identificados.
- f) Conjunto de escenarios negociados.
- g) Conjunto de preguntas para evaluación de atributos.
- h) Árbol de Utilidad.
- i) No-riesgos documentado.
- j) Puntos sensibles y de negociación.

CONCLUSIONES PARCIALES

Es necesario que el procedimiento propuesto se aplique al estilo arquitectónico SOA para ver cuán efectivo es ante este tipo de arquitecturas, y de esta manera poder analizar qué aspectos pueden ser ajustados o modificados en el mismo, de modo que se logre un mayor nivel de profundidad a la hora de aplicar dicho método.

CONCLUSIONES

- Aunque la ingeniería de software basada en componentes promete mejorar altamente los niveles de calidad de los sistemas, no deja de ser importante la evaluación de la Arquitectura de Software. Sin embargo, para poder aplicar los métodos de evaluación arquitectónica tradicionales se requiere adaptarlos a la naturaleza de los componentes para poder aplicarle el método MECABIC.
- Tanto en los métodos analizados como en el propuesto, la “técnica de escenarios” para evaluar arquitecturas es la más usada, como constatación del grado de cumplimiento de cierto atributo de calidad con los requerimientos del sistema.
- Es bueno destacar que un método de evaluación no es mejor que otro, sino que evalúa mejor, en ciertas condiciones, un atributo de calidad dado por lo que en dependencia de las condiciones y lo que se desea evaluar, será el método de evaluación empleado (que no tiene que ser solo uno).

RECOMENDACIONES

Se recomienda aplicar el método propuesto a otros estilos arquitectónicos como por el ejemplo el SOA (Arquitectura Orientada a Servicios) para ver cuán efectivo es ante otros tipos de arquitecturas, y de esta manera poder analizar qué aspectos pueden ser ajustados o modificados en el mismo, de modo que se logre un mayor nivel de profundidad a la hora de aplicar dicho método.

BIBLIOGRAFÍA

1. *Método de Evaluación de Arquitecturas de Software Basadas en Componentes (MECABIC)*. **Aleksander González, Marizé Mijares, Luis E. Mendoza, Anna Grimán, María Pérez.** Colimía, Mexico : s.n., 2005, Vol. vol 1.
2. **Erika Camacho, Fabio Cardeso, Gabriel Nuñez.** *Arquitecturas de Software*. 2004.
3. **Reinoso, Carlos Billy.** MSDN. *MSDN*. [En línea] 26 de junio de 2006. http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.msp.
4. *Evaluando Arquitecturas de Software. Parte 1. Panorama General*. **Gómez, Omar Salvador Gómez.** 01, México : Brainworx S.A, 2007. 1870-0888.
5. **Gustavo Andrés Brey, Gastón Escobar, Nicolas Passerini y Juan Arias.** *Arquitectura de Proyectos de IT. Evaluación de Arquitecturas*. Buenos Aires : Universidad Tecnológica Nacional. Facultad Regional de Buenos Aires - Departamento de Sistemas, 2005.
6. **Jorge Triñanes, María de las Nieves Freira.** Gestión de Software. *Gestión de Software*. [En línea] 2006. [Citado el: 20 de octubre de 2007.] <http://www.fing.edu.uy/inco/cursos/gestsoft/>.
7. **klein, Mark.** SEI (Software Engineering Institute). *SEI (Software Engineering Institute)*. [En línea] Carnegie Mellon University, 2007. [Citado el: 18 de noviembre de 2007.] http://www.sei.cmu.edu/architecture/ata_method.html.
8. **Clement, Paul.** SEI (Software Engineering Institute). *SEI (Software Engineering Institute)*. [En línea] agosto de 2000. [Citado el: 20 de noviembre de 2007.] <http://www.sei.cmu.edu/publications/documents/00.reports/00tn009.html>. Std. Z39-18298-102.
9. *Evaluando Arquitecturas de Software. Parte 2. Métodos de Evaluación*. **Gómez, Omar Salvador Gómez.** 02, México : Brainworx S.A, 2007. 1870-0888.
10. SEI (Software Engineering Institute). *SEI (Software Engineering Institute)*. [En línea] Carnegie Mellon University. [Citado el: 21 de noviembre de 2007.] www.sei.cmu.edu/architecture/arid.html.