

## **PROPUESTA PARA EVALUAR ARQUITECTURAS DE SOFTWARE**

### **PROPOSAL TO EVALUATE SOFTWARE ARCHITECTURES**

**Autores:** Lenna Carballo Muñoz

Ivette Barrientos Núñez

**Institución:** Universidad de Ciego de Ávila Máximo Gómez Báez

**Correo electrónico:** [lenna@unica.cu](mailto:lenna@unica.cu)

#### **RESUMEN**

La arquitectura de software ha adquirido una importancia creciente en los últimos años en el desarrollo de software a nivel internacional. Para desarrollar un producto de software dentro de los plazos de tiempo planificados, con la calidad y eficacia requerida se necesita disponer, desde etapas tempranas del proyecto una arquitectura bien definida, robusta y acorde a los factores de calidad deseados en cada caso. Actualmente en Cuba no se cuenta con métodos y procedimientos ajustados a las características de la naciente industria del software, que permitan asegurar la calidad de la arquitectura. Con la utilización de métodos teóricos, fundamentalmente analítico - sintético y empíricos se realiza el estudio resumido en el presente artículo donde se propone un procedimiento para evaluar arquitecturas de software en etapas tempranas del ciclo de vida o como también se le conoce, «Pruebas de Concepto» de la Arquitectura. La aplicación de este procedimiento pudiera contribuir a mejorar notablemente la calidad del software y a manejar los riesgos técnicos asociados al proceso de desarrollo del mismo en las empresas cubanas.

**Palabras clave:** Arquitectura de Software, Calidad, Evaluación, Pruebas de Concepto.

#### **ABSTRACT**

Software Architecture has acquired an increasing importance in software development in recent years at international level. To develop a software product within planned time

frames, with the quality and efficiency required will need to have, since early stages of project, a well-defined architecture, robust and according to the desired quality factors in each case. Nowadays there are not methods and procedures in Cuba adjusted to the characteristics of the nascent software industry. With the use of theoretical methods, mainly analytic - synthetic and empirical study was conducted as summarized in this article that proposes a method to evaluate software architectures early in the life cycle or, as is known, "Proof of Concept" of architecture. The application of this procedure could help to significantly improve software quality and manage the technical risks associated with the software development process in the Cuban enterprises.

**Keywords:** Evaluation, Proof of Concepts, Quality, Software Architecture.

## INTRODUCCIÓN

La arquitectura de software es importante como disciplina debido a que los sistemas de software crecen de forma tal que resulta muy complicado que sean diseñados, especificados y entendidos por un solo individuo. Uno de los aspectos que motivan el estudio en este campo es el factor humano, en términos de aspectos como inspecciones de diseño, comunicación a alto nivel entre los miembros del equipo de desarrollo, reutilización de componentes y comparación a alto nivel de diseños alternativos. (Kazman, 1996).

La Arquitectura de Software es una disciplina relativamente joven, sus orígenes datan de los años '60 pero no es hasta la década del 90 en que se define como hoy la conocemos y se comienza a realizar un trabajo intensivo alrededor de ella. La década del 90 se considera como la década de la arquitectura de software según algunos principales exponentes como Dewayne Perry y Alexander Wolf.

La arquitectura de software (AS) de los sistemas de software (SS) a ser construidos se convierte en un factor de importancia para lograr que este tenga un alto nivel de calidad. Recuérdese que poseer una buena AS es de suma importancia ya que esta es el corazón de todo SS y determina cuáles serán los niveles de calidad asociados al sistema. (Alexander, G., 2005).

Cuando no se analiza rigurosamente la calidad de la arquitectura antes de la construcción a gran escala del sistema pueden presentarse problemas. Mientras más tarde se detecten estos, mayor será el impacto que tendrán. Se considera que el 50% de los proyectos atrasados y el 35% de los excedidos en costo de producción tienen problemas en la arquitectura.

Principales problemas causados por una arquitectura defectuosa en un proyecto de desarrollo de software:

- ✓ Incumplimiento de los niveles de calidad propuestos.
- ✓ Incongruencia entre las necesidades del negocio y el proyecto en proceso de desarrollo.
- ✓ Falta de compromiso del usuario en la culminación del proyecto (problemas de comunicación).
- ✓ Atraso en el proceso de producción del software.
- ✓ Aumento de los costos durante el proceso de producción del software.

Para disminuir los riesgos asociados a la arquitectura se hace necesario evaluar la calidad de esta antes de la construcción del sistema, lo que comúnmente se denomina como Pruebas de Concepto de la Arquitectura.

Evaluar una Arquitectura no es fácil, a menudo priman muchos factores empíricos y es difícil, sobre todo cuantificar cuanto satisface o no a los requerimientos del sistema. En los últimos años han surgido varios métodos y técnicas con el propósito de facilitar esta tarea, ejemplo de ello son: Método ATAM, Método QUASAR, Método MECABIC, además de listas de chequeo. Todos ellos se valoran en el presente trabajo con el propósito de extraer los elementos que sean apropiados a nuestro entorno y dar como propuesta un procedimiento que permita realizar una evaluación de calidad a la arquitectura de software con el fin de minimizar los riesgos asociados a esta, así como aplicarlo a los nuevos proyectos de desarrollo de software a que se enfrenta la Facultad Regional de Ciego de Ávila perteneciente a la Universidad de las Ciencias Informáticas (FRCA UCI).

La investigación se enmarca dentro del área de Gestión de la Calidad expresada en

PMBOK 2004, esta área comprende Planificación, Aseguramiento y Control de la Calidad, la solución propuesta está comprendida mayormente dentro de las actividades de Aseguramiento de la Calidad del Software.

## **MATERIALES Y MÉTODOS**

### Principales Conceptos

#### Arquitectura de Software

Existe un número bastante elevado de definiciones de Arquitectura de Software lo cual alcanza un orden de tres dígitos, aquí se exponen algunas de las más reconocidas por instituciones internacionales.

Una definición reconocida es la de Clements (1996):

«La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión del detalle inherente a la mayor parte de las abstracciones».

#### Calidad de Software

En la mayoría de las definiciones se resalta el hecho de que un software de calidad debe satisfacer los requerimientos dados por el usuario.

La IEEE, citado por (Barbacci, et al., 1995) afirma que «la calidad de un software es el grado en el cual el software posee una combinación deseada de factores». Este concepto quizás esté motivado por el hecho de que los atributos de calidad de un software pueden en cierta medida entrar en contradicción, por ejemplo para ganar en Seguridad podemos perder en cuanto a Rendimiento, por lo cual es necesario para un software tener no solo bien detallados los requerimientos funcionales y no funcionales, sino también sus prioridades en cuanto a los atributos, esto debe formar parte de la concepción arquitectónica.

Sin un proceso sistemático que permita garantizar la calidad dentro de un proceso de desarrollo de software no tendría ninguna importancia hablar de Calidad de Software, por eso es vital tratar el tema de Aseguramiento de la Calidad de Software.

### Aseguramiento de la Calidad de Software

Denominado en la mayoría de las empresas como SQA o Software Quality Assurance, permite elaborar actividades sistemáticas que se necesitan para lograr la calidad en el producto, que en este caso es un software (Pressman, 2002). También (Pressman, 2002) afirma que el Aseguramiento de la Calidad puede tener las siguientes actividades: evaluaciones en las etapas del desarrollo, auditorías y revisiones, estándares que se aplicarían al proyecto, mecanismos de medida (métricas), métodos y herramientas de análisis, diseño, programación y prueba, documentación y control de software. El Aseguramiento de la Calidad propicia ciertos atributos y características que influyen significativamente en la Calidad de Software.

### Evaluación

Es un estudio de factibilidad que pretende detectar posibles riesgos, como así también buscar recomendaciones para contenerlos.

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.

### Escenarios

Un escenario es una breve descripción de la interacción de alguno de los involucrados en el desarrollo del sistema. Este consta de tres partes: el estímulo, el contexto y la respuesta (Kazman, 1996).

### Decisión arquitectónica

Las dediciones arquitectónicas son aquellas que decide tomar el arquitecto que tienen un significado desde el punto de vista de la arquitectura.

### Importancia de la evaluación de AS

La evaluación de una arquitectura de software pretende medir propiedades del sistema en base a especificaciones abstractas como son 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. El propósito de realizar evaluaciones a la

arquitectura, es fundamentalmente para analizar e identificar riesgos potenciales en su estructura y sus propiedades que puedan afectar al sistema de software resultante, verificar que los requisitos no funcionales estén presentes en la arquitectura, así como determinar en que grado se satisfacen los atributos de calidad. Cabe señalar que los requisitos no funcionales también son llamados atributos de calidad.

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.

(Kazman, et. al, 1996) proponen un esquema general en relación a la evaluación de una arquitectura con respecto a sus atributos de calidad. 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.

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.

Bosch plantea las técnicas de evaluación: basada en escenarios, en simulación, en modelos matemáticos y 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, dado que estos parámetros son, por lo general, dependientes de la implementación. Su enfoque se orienta hacia la mitigación de riesgos, ubicando donde un atributo de calidad de interés se ve afectado por decisiones arquitectónicas.

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 entonces definir los atributos de calidad en función de sus metas y su contexto y no como cantidades absolutas.

¿Cómo puedo estar seguro de 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.

¿Cuándo una arquitectura puede ser evaluada?

Es posible realizar la evaluación de la arquitectura en cualquier momento, según (Kazman, 1996) pero propone dos variantes que agrupan dos etapas distintas: Temprana y Tardía.

Temprana: No es necesario que la arquitectura esté completamente especificada para efectuar la evaluación y esto abarca las fases tempranas del diseño y a lo largo del desarrollo.

Tardía: Cuando esta 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 desarrollado.

Bredemeyer propone que los requerimientos arquitectónicos son necesarios para guiar las actividades de estructuración de un sistema de software. En el proceso de desarrollo se pueden aplicar diversos enfoques para garantizar el cumplimiento de los requerimientos arquitectónicos, así como la evaluación de las alternativas presentadas. La evaluación provee indicadores que permiten en las fases tempranas, la oportunidad de resolver problemas que pueden presentarse a nivel arquitectónico.

Independientemente de la metodología implementada, la intención es obtener una arquitectura con la documentación necesaria y asegurar que el sistema cumple con los servicios y la funcionalidad que espera el usuario, además de los atributos de calidad asociados que deben cumplirse y que dirigen las decisiones al momento de la construcción de la arquitectura del sistema. (Bredemeyer, 2002).

## Métodos y modelos existentes para evaluar la calidad de la arquitectura

### Software Architecture Analysis Method (SAAM)

El método de evaluación SAAM evalúa distintos atributos de calidad como modificabilidad, portabilidad, escalabilidad e integrabilidad. Este se enfoca en la enumeración de un conjunto de escenarios que representan los cambios probables a los que estará sometido el sistema en el futuro. Como entrada principal, es necesaria alguna forma de descripción de la arquitectura a ser evaluada. Las salidas de la evaluación del método SAAM son las siguientes:

- ✓ Una proyección sobre la arquitectura de los escenarios que representan los cambios posibles ante los que puede estar expuesto el sistema.
- ✓ Entendimiento de la funcionalidad del sistema, e incluso una comparación de múltiples arquitecturas con respecto al nivel de funcionalidad que cada una soporta sin modificación.

Con la aplicación de este método, si el objetivo de la evaluación es una sola arquitectura, se obtienen los puntos en los que la misma puede fallar, en términos de los requerimientos de modificabilidad. Para el caso en el que se cuenta con varias arquitecturas candidatas, el método produce una escala relativa que permite observar qué opción satisface mejor los requerimientos de calidad con la menor cantidad de modificaciones.

### Architecture Trade-off Analysis Method (ATAM)

Apoya a los involucrados con el proyecto a entender las consecuencias de las decisiones arquitectónicas a los atributos de calidad (requisitos no funcionales como: seguridad, disponibilidad, desempeño (performance), facilidad de mantenimiento) del sistema.

### Active Reviews for Intermediate Designs (ARID)

Según (Kazman, 1996) el método ARID es conveniente para realizar la evaluación de diseños parciales en las etapas tempranas del desarrollo. En ocasiones, es necesario saber si un diseño propuesto es conveniente, desde el punto de vista de otras partes de la arquitectura. Según el autor, ARID es un híbrido entre Active Design Review (ADR) y Architecture Trade-Off Method (ATAM), descrito anteriormente.

ADR es utilizado para la evaluación de diseños detallados de unidades del software como los componentes o módulos.

### MECABIC

El MECABIC está inspirado en ATAM, aunque con el objetivo de facilitar su aplicación sobre las arquitecturas basadas en componentes 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). (Bachmann, et. al., 2000).

Además se propone un árbol de utilidad inicial basado en el modelo de calidad ISO-9126 instanciado para AS propuesto por Losavio (Practice for Architectural Description of Software-Intensive System, 2000). La adopción de este modelo por parte del MECABIC permite concentrarse en características que dependen exclusivamente de la arquitectura.

### Método de Bosch

Método Diseño y Uso de Arquitecturas de Software propuesto por (Bosch, 2000) plantea que el proceso de evaluación debe ser visto como una actividad interactiva, que forma parte del proceso de diseño, también interactivo. 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. El proceso de evaluación propuesto por (Bosch, 2000) se divide en dos etapas:

#### Etapa I

1. Selección de atributos de calidad
2. Definición de los perfiles
3. Selección de una técnica de evaluación

#### Etapa II

4. Ejecución de la evaluación
5. Obtención de resultados

El análisis realizado a cada uno de los métodos anteriores de evaluación de arquitecturas permitió identificar áreas comunes entre estos, como lo es el uso de

escenarios como técnica de evaluación, la cual se encontró en algunos casos combinada con otras técnicas generando una evaluación más efectiva. En estos métodos se refleja la importancia de la especificación exhaustiva de los atributos de calidad como base para efectos de la evaluación de una arquitectura de software, por lo que la mayoría de ellos se enfocan en diferentes atributos de calidad para realizar la evaluación, habiendo otros que dejan la identificación de estos atributos por parte de los arquitectos según la importancia sobre el sistema.

## **RESULTADOS Y DISCUSIÓN**

La FRCA UCI, con vista a los nuevos proyectos de desarrollo de software a los que se enfrentará, tendría la necesidad de minimizar en todos ellos los riesgos asociados a la arquitectura y a su vez es necesario que el procedimiento utilizado para esto no sea muy costoso, ni burocrático, puesto que con la gran cantidad de tareas que se generan normalmente en un proyecto la salida más común será no realizarlo.

En la Fig.1 se propone un procedimiento general que trata de ajustarse a las restricciones antes mencionadas y parte de las estructuras productivas ya establecidas en la organización y que tiene como base los elementos comunes de los métodos de evaluación anteriormente analizados.

El procedimiento tal como se describe, no pretende ser exhaustivo sino servir de guía elemental de las actividades que se deben realizar para evaluar la arquitectura de software en una etapa temprana del proyecto. Propone algunas técnicas a utilizar, aspectos a tener en cuenta así como los involucrados en cada una de las actividades que se realizan.

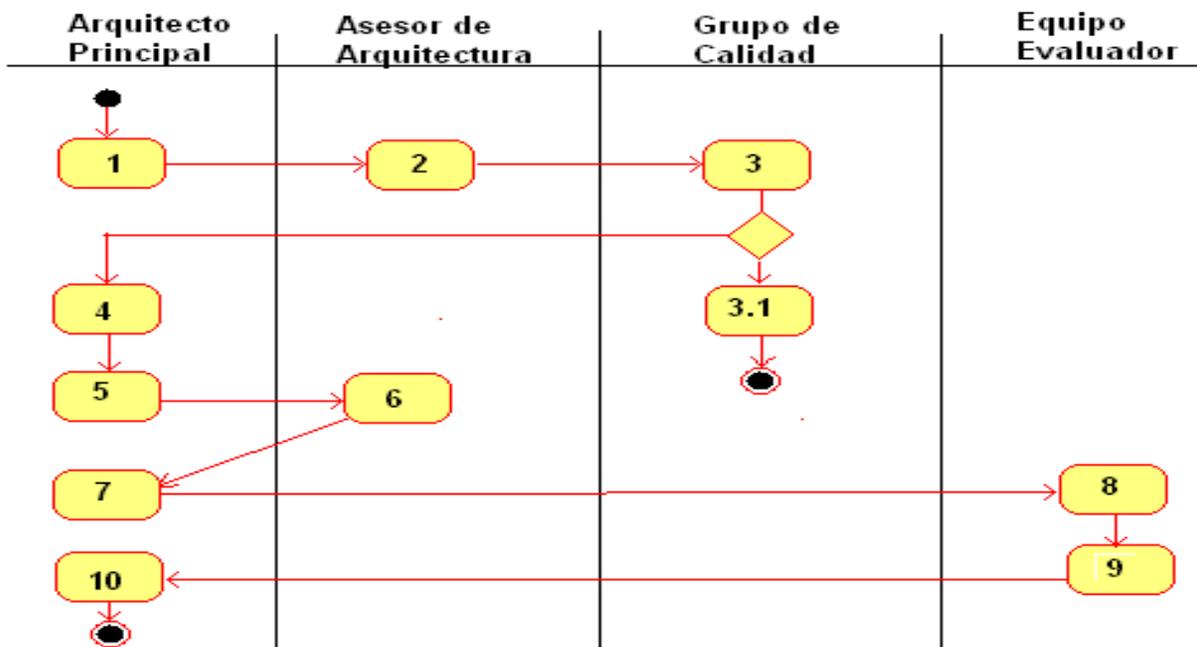


Fig. 1: Propuesta para evaluar Arquitecturas de Software

**Paso 1:** Solicitar la evaluación de la arquitectura de su proyecto.

Debe estar establecido por la institución que la arquitectura de todos los sistemas debe someterse a un proceso de pruebas de concepto antes de su implementación a gran escala. Luego de que el arquitecto tenga una propuesta inicial de arquitectura debe solicitar que se inicie el proceso de evaluación.

**Paso 2:** Recopilar la documentación y enviarla al grupo de calidad.

El asesor debe revisar la propuesta inicial y decidir si tiene condiciones para someterse al proceso de evaluación. Obtiene la documentación completa y gestiona el inicio del proceso, convoca al equipo de calidad y convoca al grupo de expertos que efectuarán la evaluación. Es imprescindible que los miembros del equipo de evaluación tengan conocimientos sobre arquitectura y preferiblemente que posean experiencia como arquitectos en el tipo de sistema que se evalúa.

**Paso 3:** Determinar si tiene condiciones mínimas para comenzar la evaluación.

Para transitar del paso 3 al paso 4 se necesita un nivel mínimo de madurez de la arquitectura antes de que se le pueda aplicar exitosamente el proceso de evaluación.

Algunos de los elementos a verificar antes de comenzar son:

- ✓ Los requerimientos o atributos de calidad han sido correctamente derivados y especificados de forma que sean medibles o verificables.
- ✓ La arquitectura ha sido documentada apropiadamente en términos de diagramas, modelos y documentos cumpliendo con las normas de calidad establecidas en el Proyecto, Polo Productivo o Institución.
- ✓ La arquitectura tiene un nivel de especificación que hace factible someterla al proceso de evaluación.

**Paso 3.1:** No se aprueba la arquitectura.

Se comunican al equipo de proyecto las razones de la No Aprobación

**Paso 4:** Prepara los escenarios para los atributos de calidad prioritarios.

Algunos elementos que debe incluir el arquitecto en esta presentación son:

- ✓ Las funcionalidades más importantes del sistema objetivo del negocio o contexto dentro del cual va a funcionar el sistema.
- ✓ Directrices arquitectónicas o requerimientos de calidad a ser satisfechos por la arquitectura (restricciones técnicas, gerenciales, económicas, políticas y límites del sistema).
- ✓ Estructura lógica y relaciones entre componentes de la arquitectura,
- ✓ Decisiones arquitectónicas tanto de diseño (patrones) como restricciones técnicas (sistema operativo, hardware, otros sistemas con los que debe interactuar).
- ✓ Decisiones arquitectónicas que aún no han sido tomadas.
- ✓ Escenarios más relevantes para los atributos de calidad que guían la arquitectura.

**Paso 5:** Identificar riesgos y su impacto en la arquitectura.

El arquitecto debe evaluar los riesgos asociados a cada escenario, para ello se puede auxiliar de la Tabla 1:

Escenario	Atributo de calidad que afecta	de que decisiones arquitectónicas que involucra	Riesgos para el sistema	Impacto (alto, medio, bajo)

*Tabla 1: Aspectos a tener en cuenta en la evaluación.*

**Paso 6:** Convocar a la reunión de evaluación.

El asesor de arquitectura entrega al equipo evaluador la documentación preparada por el arquitecto.

El equipo evaluador la revisa, puede hacer sugerencias o pedir al arquitecto que incluya el análisis de atributos de calidad relevantes para la arquitectura que se analiza y escenarios asociados que no fueron considerados. Cuando la documentación está lista, el equipo evaluador lo comunica al asesor y este convoca a la reunión de evaluación.

**Pasos 7 y 8:** Efectúa la evaluación.

En estos dos pasos se efectúa la evaluación donde el arquitecto expone los puntos abordados en el paso 4. El equipo de evaluadores debe señalar los puntos no esclarecidos hasta que todos los puntos de la exposición sean entendidos de ambas partes.

**Paso 9:** Se reúne el equipo evaluador.

Concluida la reunión, el equipo evaluador se reúne, valora las decisiones arquitectónicas (puede utilizar método de expertos) y elabora la documentación final donde debe incluir:

- ✓ Riesgos no considerados para las decisiones arquitectónicas, así como su posible impacto.
- ✓ Atributos de calidad significativos que se deberían considerar.

- ✓ Se podrán proponer decisiones arquitectónicas más efectivas que las propuestas inicialmente, siempre que se justifique como evitarían o mitigarían algún riesgo, valorando el esfuerzo de llevar a cabo las ventajas de su ejecución.
- ✓ Decisiones arquitectónicas que es necesario cambiar por sus consecuencias para el proyecto o la institución.

En caso de algún conflicto de criterios entre la parte evaluada y la evaluadora debe ser negociado por medio del asesor. Se debe concluir con la respuesta del arquitecto sobre los aspectos que serán modificados a partir de los señalamientos y recomendaciones hechas. El equipo evaluador puede determinar si la arquitectura deberá ser sometida a otra evaluación de tipo conceptual antes de ser implementada.

#### Paso 10: Llevar a cabo los cambios.

Se llevan a cabo los cambios por el arquitecto y equipo de desarrollo.

Los objetivos que se deben lograr al terminar el proceso son:

- ✓ Identificar la mayoría de los riesgos asociados a la arquitectura (en una etapa temprana en que todavía pueden ser mitigados), así como sus fortalezas acorde a la duración y profundidad de la evaluación.
- ✓ Obtener un estimado inicial de la magnitud, probabilidad de ocurrencia e impacto de la ocurrencia de los riesgos arquitectónicos identificados.
- ✓ Identificar los requerimientos de calidad que no han sido tenidos en cuenta o que no son soportados adecuadamente por la arquitectura.

#### Valoración económica y aporte social

Dada la importancia que se le atribuye a nivel mundial a la arquitectura dentro del proceso de desarrollo de un software, además de la poca experiencia que existe en este campo en la FRCA UCI y con vista a los nuevos proyectos de desarrollo que se enfrentan en el centro, se pretende que la presente propuesta contribuya a la calidad final del software desarrollado y que mediante su implantación se convierta en un proceso clave, dando esto como resultado un mejor orden y seguridad.

A partir de este modelo se lograría una arquitectura de calidad desde las primeras fases de su concepción aportando valor añadido en materia de eficiencia y eficacia.

## CONCLUSIONES

La evaluación de la arquitectura de software en una etapa temprana del proyecto constituye una necesidad y un medio efectivo para mitigar los riesgos asociados a la arquitectura y con ello minimizar su impacto negativo. El procedimiento propuesto se basa en los aspectos más recurrentes y significativos de un conjunto de métodos propuestos por otros autores, constituye una base para llevar a cabo dicha evaluación de forma que no sea muy costosa en cuanto a tiempo y recursos del proyecto, es adaptable a las características de la arquitectura y de la organización.

## BIBLIOGRAFÍA CONSULTADA

BACHMANN, F.B.L.: *The Architectural Based Design Method*. Disponible en [www.sei.cmu.edu/pub/documents/](http://www.sei.cmu.edu/pub/documents/). Visitado el 14 de marzo de 2014.

BARBACCI, M.K.M.; LONGSTAFF, T. Y WEINSTOCK, C.: *Quality Attributes*, Carnegie Mellon University. Disponible en [www.sei.cmu.edu/publications/documents/95.reports/95.tr.021.html](http://www.sei.cmu.edu/publications/documents/95.reports/95.tr.021.html). Visitado el 14 de marzo de 2014

BOSCH, J.: *Design & Use of Software Architectures*, Ed. Addison Wesley, 2000.

BREDEMEYER, D.M.: *The Visual Architecting Process*. Disponible en [www.bredmeyer.com/pdf files/](http://www.bredmeyer.com/pdf files/). Visitado el 14 de marzo de 2014.

CLEMENTS, P.: *A Survey of Architecture Description Languages*, Software Engineering Institute, Carnegie Mellon University. Disponible en [www.sei.cmu.edu/publications/articles/survey-adl.html](http://www.sei.cmu.edu/publications/articles/survey-adl.html). Visitado el 18 de marzo de 2014.

ALEXANDER, M.M.: *Método de Evaluación de arquitecturas de Software Basadas en Componentes (MECABIT)*, 2005.

KAZMAN, R.: *Tool Suport for Architecture Analysis and Design*, Department of Computer Sciencie, University of Waterloo. Disponible en <ftp.sei.cmu.edu/pub/sati//Papers>.

Visitado el 19 de marzo de 2015.

LOSAVIO, F.C.; LÉVY, L. Y RANDANE, CH.: *A quality Caharakteristica for Software Architecture*, 2003.

PMBOK.: *Guía de los fundamentos de la Dirección de Proyectos (Guía del PMBok)*, Campus Boulevard, Newtown Square EE.UU, Project Managment Institute, 2004.

Practice for Architectural Description of Software-Intensive System. IEEE, 2000.

PRESSMAN, R.: *Ingeniería de Software, un Enfoque Práctico*. Mc Graw Hill, 5ta Edición, 2002.