



Universidad de las Ciencias Informáticas

**TESIS EN OPCIÓN AL TÍTULO ACADÉMICO DE
MASTER EN GESTIÓN DE PROYECTOS
INFORMÁTICOS**

**Metodología de evaluación de
arquitecturas de software**

Autor: Ing. Yamila Vigil Regalado

Tutor: Dr. Ailyn Febles Estrada

La Habana, 2009.

“Año 50 de la Revolución”

Resumen.

Elevar la calidad del producto final y del proceso de desarrollo del software constituye la meta principal de la industria de producción de software en la actualidad. Este objetivo contribuye no solo a lograr mejor competitividad antes las altas exigencias de los mercados, sino que ocasiona la disminución de esfuerzo y tiempo en los desarrollos, que proporcionalmente provoca una reducción considerable de los costos de cada producto.

Debido al aumento de la complejidad de los sistemas informáticos, el enfoque de la arquitectura de software ha ocupado en la última década un lugar significativo en el ciclo de desarrollo de un producto.

La evaluación de la arquitectura de software consiste en la verificación de los factores de calidad desde la perspectiva del diseño arquitectónico. Además de identificar los puntos críticos que presenta la arquitectura del sistema en lograr estos factores y las exigencias de los clientes. Es por ellos que evaluar este artefacto que se genera durante el ciclo de desarrollo es vital para aumentar la calidad.

Este trabajo propone una metodología que instrumenta el proceso de evaluación de arquitectura, un conjunto de acciones a seguir para lograr una aplicación exitosa de la misma y un conjunto de plantillas que facilitan el proceso de documentación de la evaluación arquitectónica.

El uso y aplicación de la propuesta contribuirá a la disminución de esfuerzo, tiempo y costo en proyectos productivos de la universidad, así como el aumento de la calidad del proceso de desarrollo y del producto final.

Índice

Introducción	6
Diseño teórico de la investigación.....	10
Diseño metodológico de la investigación.	13
Estructura de este documento.....	15
Capítulo 1. Introducción a la evaluación de las arquitecturas de software.....	17
1.1 Introducción.....	17
1.2 La arquitectura de software: evolución y definición de un concepto.	17
1.3 Calidad arquitectónica.....	20
1.3.1 Modelos de calidad.....	22
1.3.1.1 Modelo de McCall.....	23
1.3.1.2 Modelo de Boehm.	26
1.3.1.3 Modelo de Dromey	28
1.3.1.4 Modelo FURPS	30
1.3.1.5 Modelo ISO/IEC 9126.....	31
1.3.1.6 Consideraciones generales sobre los modelos de calidad.	41
1.4 Evaluación de arquitecturas de software.....	42
1.4.2.1 Método de análisis de arquitectura de software (Software Architecture Analysis Method, SAAM).....	50
1.4.2.2 Método de análisis de interacciones de la arquitectura de software. (Architecture Trade-off Analysis Method, ATAM)	55
1.4.2.3 Método de análisis de costo-beneficio. (Cost Benefit Analysis Method, CBAM) 60	
1.4.2.4 Método de análisis del nivel de modificación de la arquitectura. (Architecture Level Modifiability Analysis, ALMA).....	64

1.4.2.5	Método de análisis familiar de la arquitectura. (Family – Architecture Analysis Method, FAAM).....	70
1.4.2.6	Consideraciones generales sobre los métodos de evaluación de arquitectura. 76	
1.5	Experiencia internacional	76
1.6	Conclusiones del capítulo.....	78
Capítulo 2:	Metodología de evaluación de arquitectura de software.	79
2.3.1	Entrevista	83
2.3.2	Encuesta.....	84
2.3.3	Revisión documental.	85
Capítulo 3.	Proceso de implementación de la metodología.	110
3.1	Introducción.....	110
3.2	Proceso de implementación de la metodología.	110
3.2.1	Acciones que garantizan el nivel de partida de la evaluación.	111
3.2.2	Acciones que facilitan el proceso de evaluación y garantizan la calidad del mismo. 111	
3.2.3	Acciones que garantizan replicar experiencias en proyecto similares.	113
3.3	Análisis de los resultados de la aplicación de las encuestas y entrevistas.	113
3.4	Análisis de los resultados de evaluación técnica del modelo.	119
3.5	Análisis de los resultados de la utilización de la metodología en un proyecto piloto. 121	
3.6	Conclusiones del capítulo.....	127
Conclusiones y recomendaciones.....		128
Conclusiones.		128
Recomendaciones.		129
Bibliografía		130
Anexos.		133
Anexo 1. Diseño de la entrevista realizada.		133

Anexo 2. Diseño de la encuesta # 1.....	134
Anexo 3. Diseño de la encuesta # 2.....	136
Anexo 4. Modelos y tablas para la evaluación técnica.....	137
Anexo 5. Plantillas propuestas para la evaluación de la arquitectura de software.....	144
Anexo 6. Escenarios identificados durante le proceso de evaluación.	179

Introducción

Durante la última década, la funcionalidad de los modernos sistemas electrónicos ha aumentado drásticamente: los televisores pueden ser utilizados para la comunicación por Internet, la telefonía móvil permite grabar y reproducir video, los sistemas de imágenes médicas puede realizar la reconstrucción 3D de los vasos sanguíneos en tiempo real; y para todos estos sistemas, es su software interno quien proporciona el valor añadido.

El papel de los programas informáticos se ha convertido en esencial, permitiendo la aplicación de la diversidad y el aumento de la funcionalidad de los productos de hardware. En muchos casos, la aplicación de la funcionalidad del software logra un uso y aprovechamiento del hardware a menores costos. Por otra parte, en comparación con el hardware, el software es fácilmente modificable, actualizable y personalizable. Las mencionadas características del software conllevan a un aumento significativo del margen de beneficio del producto.

Cada día la complejidad y el tamaño del software van en ascenso, a la par de las funcionalidades que debe lograr el mismo. Paralelo a este suceso, las empresas de desarrollo de productos de software están dispuestas a seguir siendo competitivas en el mercado, y para esto deben lograr:

- a) reducir el tiempo de comercialización,
- b) reducir los costes de producción,
- c) diversificar los productos para cubrir en lo posible los sectores del mercado,
- d) elevar la calidad de los productos que desarrollan.

Estos pasos pueden ayudar a las empresas a tener éxito en el mercado, aumentar sus beneficios, y poder reaccionar a los cambios en el mundo de los negocios de una manera más flexible.

Para lograr este objetivo, no sólo la funcionalidad del software desempeña su papel, sino también otras cualidades como la mantenibilidad del software, la fiabilidad, la facilidad de uso, entre otros, que definen cómo el software puede abordar los distintos aspectos de su contexto con la calidad requerida.

De acuerdo con la norma ISO 9126-1 (ISO/IEC-9126-1, 2001) la calidad se define como un conjunto de rasgos y características de un producto o servicio que le confieren su aptitud para satisfacer necesidades explícitas o implícitas. Diferentes perspectivas de la calidad se pueden considerar:

- ✓ La opinión del usuario, como la calidad del producto final.
- ✓ La vista del desarrollador, como la calidad de los productos intermedios, generados en las diferentes etapas durante el proceso de desarrollo.
- ✓ La vista del administrador de usuarios finales, como la calidad de los requisitos de comercialización.

La calidad general del producto puede ser expresada por una combinación de los diferentes puntos de vista mencionados.

La obra de McCall establece dos niveles de características de la calidad: factores y criterios. El primero no se puede medir directamente, mientras que el segundo puede ser medido subjetivamente. Este último inspiró el modelo de la ISO 9126-1. Sobre esta base, la norma ISO 9126-1 simplifica aún más el modelo de McCall y es aceptado comúnmente como técnica en la especificación de la calidad del producto a nivel internacional.

Esta norma propone un conjunto de seis características de calidad de alto nivel, que se definen como un conjunto de atributos de un producto de software por el cual su calidad es descrita y evaluada. Los atributos de calidad se utilizan como los objetivos para la *validación* externa de la calidad y la *verificación* de la calidad interna en las diferentes etapas de desarrollo. Estos son refinados en subcaracterísticas, hasta que se obtenga los atributos o propiedades medibles. En este contexto, la métrica o medida es definida como un método de medición, que permite utilizar un indicador o una medida para asignar un valor.

Con el fin de supervisar y controlar la calidad de software durante el proceso de desarrollo, los requisitos de calidad externos, se traducen o se transfieren a los requisitos de los productos intermedios, obtenidos de las actividades de desarrollo. La traducción y la selección de los atributos es una actividad no trivial, en función de la experiencia personal de los interesados, a menos que la organización proporcione una infraestructura para recoger y analizar la experiencia previa en proyectos terminados.

Durante el desarrollo de este trabajo se profundizará en el logro de la calidad según el punto de vista del desarrollador, o sea en el logro de la calidad de los productos intermedios que se generan durante el ciclo de vida de un software, en especial en la arquitectura de un sistema.

Con el fin de evaluar la calidad arquitectónica, se debe tener conciencia de las propiedades que se esperan de la arquitectura o marco genérico (línea base) en la que el sistema de software debe ser construido. Por lo tanto una arquitectura determinada, expresada como una vista de alto nivel de los componentes, las interconexiones entre ellos y la configuración o topología del sistema, debe satisfacer las seis características de la norma ISO 9126 o un subconjunto de ellos.

Sin embargo, el proceso de evaluación arquitectónica no solo debe abarcar el modelo de calidad a seguir para la verificación de los requerimientos de calidad, sino que va acompañado de un grupo de acciones organizativas, de exposición de enfoques arquitectónicos, y de los intereses de cada grupo que participa, en especial arquitectos-desarrolladores y clientes-usuarios finales.

Un paso importante en la valoración arquitectónica es priorizar los requerimientos de calidad, porque en muchas ocasiones el modelo de calidad no permite ser definitorios sobre una arquitectura candidata u otra, debido a que la verificación de los atributos muestra la misma cantidad en un enfoque arquitectónico como en otro, y es ahí donde el equipo de evaluación debe dar la prioridad de los atributos indispensables a cumplir en función de las características del producto que se desarrolla.

La arquitectura de software es considerado en la actualidad uno de los productos intermedios generados en el ciclo de desarrollo que alcanza mayor importancia e impacta en la calidad final del producto, así como en el futuro proceso de desarrollo. La arquitectura de un sistema se puede definir como:

...“la estructura o estructuras del sistema, que incluyen los componentes de software, la visibilidad exterior de las propiedades de los componentes y las relaciones entre ellos.”

Luego que el diseño arquitectónico ha sido definido este proporciona a los desarrolladores una visión clara sobre las funciones, responsabilidades, dependencias y las interacciones entre los componentes que conforman el sistema; ayuda a los integrantes de los equipos de software en la realización de los

componentes que interactúan con el medio en que se desarrolla la aplicación, y contribuye en gran medida a lograr agilidad y claridad en el proceso de desarrollo lo que provoca un alto impacto en la reducción de tiempo, costo y esfuerzo en cada uno de los proyectos.

Muchas han sido las investigaciones que han tenido como centro la evaluación de la arquitectura de software. Existen un conjunto de técnicas que evalúan el cumplimiento de los requisitos de calidad, las basadas en escenarios, las basadas en modelos matemáticos, las basadas en simulación y las que se basan en experiencia. A su vez un grupo de métodos que tienen como objetivo la evaluación de la calidad arquitectónica, entre los que se pueden mencionar, el Software Architecture Analysis Method (SAAM), el Architecture Trade-off Analysis Method (ATAM), el Active Reviews for Intermediate Designs (ARID), el Cost Benefit Analysis Method (CBAM), el Architecture Level Modifiability Analysis (ALMA), el Family – Architecture Analysis Method (FAAM), entre otros. (Bosch, 2000) (Kazman, y otros, 2003)

El ATAM que constituye uno de los enfoques más promovidos, utiliza un nivel de refinamiento llamado árbol de utilidad, que expresa las características de calidad a cumplir. Sin embargo es cuestionable porqué usar un único nivel de refinamiento y como la definición de los atributos no están normada con claridad. El árbol de utilidad se utiliza para dar prioridades a los escenarios en la identificación de puntos sensibles. Los escenarios son definidos basados en un conjunto de casos de pruebas derivados de la propia arquitectura definida. Estos atributos que plantea el método no responden ni están normados internacionalmente en un modelo de calidad, y la medida de los atributos son muy generales, poco apropiado para su aplicación en un sistema con un contexto específico.

La industria cubana del software avanza hacia su perfeccionamiento. La madurez alcanzada por el país en desarrollos con una perspectiva industrial, está dando sus primeros pasos. Es por esto que el aumento de la calidad de los productos cubanos de software es de vital importancia en el logro de una posición privilegiada en el mercado internacional.

La Universidad de las Ciencias Informáticas (UCI) es la primera universidad cubana surgida al calor de la batalla de ideas, sobre la base del nuevo concepto de universidad productiva, logrando una fuerte vinculación Estudio-Trabajo y

Universidad-Industria. Es un objetivo fundamental de la producción de software en la UCI, alcanzar en sus productos un marcado matiz industrial, que permita el aumento de la calidad de los productos y la reducción sustancial de tiempo, costo y esfuerzo. Es por ello que la evaluación arquitectónica, fundamentalmente en las etapas tempranas, constituye una práctica indispensable en sus desarrollos.

Sin embargo, luego de aplicar un grupo de encuestas, se determinó que el 78 % de los proyectos de producción de software de la universidad, no llevan a cabo la actividad de evaluación de la arquitectura, lo que ha conllevado problemas como:

- ✓ Decisiones arquitectónicas que no responden al logro de los atributos de calidad esperados.
- ✓ Demoras en la entrega de desarrollo debido a deficiencias arquitectónicas detectadas durante la implementación.
- ✓ Imposibilidad en etapas tempranas de tomar decisiones de diseño, capaces de reducir tiempo, costo y esfuerzo en las venideras fases del ciclo de desarrollo.
- ✓ No documentación del proceso de evaluación arquitectónica, verificación de los requerimientos de calidad, y decisiones arquitectónicas que arrojaron los procesos evaluativos, que permitan replicar experiencias en otros proyectos desarrollados por la organización.

Esta investigación se basa fundamentalmente en esta problemática, y persigue brindar una solución que pueda ser generalizable a los proyectos de desarrollo de la universidad.

Diseño teórico de la investigación.

Resumiendo los elementos mencionados anteriormente se puede concluir lo siguiente:

- ✓ El aseguramiento de la calidad durante el proceso de desarrollo del software es de suma importancia pues permite un aumento significativo de la calidad del producto final, a la par que reduce costos, esfuerzo y tiempo.
- ✓ La arquitectura de software es un producto intermedio que se genera durante el ciclo de desarrollo de un producto.

- ✓ Que la arquitectura de software propuesta logre el cumplimiento de los atributos necesarios para el logro de la calidad interna del producto, es significativo para el buen cumplimiento de las demás fases del proceso de desarrollo y para el logro de la calidad final del producto.

Partiendo de la necesidad de lograr altos niveles de calidad en cada uno de los productos que desarrolla la UCI, así como que la arquitectura definida en cada proyecto cumpla con los requerimientos de calidad y sirva de línea base para las restantes etapas del desarrollo, brindando agilidad y claridad a los desarrolladores, se arriba al siguiente *Problema de investigación*:

¿Cómo realizar el proceso de evaluación de la arquitectura de software en el desarrollo de un producto informático?

A partir de este problema se definió como *objeto de estudio* el proceso de evaluación arquitectónica como factor de impacto en la calidad del proceso de desarrollo y del producto final y como *campo de acción* el proceso de evaluación de la arquitectura de software en los proyectos productivos de la UCI.

Para responder a la interrogante anteriormente expuesta se trazó el siguiente *Objetivo general*:

Diseñar una metodología que permita realizar una evaluación integral de las arquitecturas de software diseñadas en los proyectos de desarrollo de software en la UCI.

A partir del análisis del objetivo general se derivan los siguientes *Objetivos específicos*:

- ✓ Elaborar el marco teórico con relación a la arquitectura de software y dejar definida la posición del investigador.
- ✓ Caracterizar el proceso de evaluación de arquitectura de software.
- ✓ Analizar y caracterizar las herramientas de evaluación de arquitectura existentes.
- ✓ Proponer la utilización de un modelo de calidad en el proceso de evaluación de la arquitectura.
- ✓ Identificar los problemas existentes en la producción de software de la UCI.

- ✓ Proponer y evaluar técnicamente una metodología de evaluación de arquitectura para los proyectos productivos de la UCI.
- ✓ Diseñar y ejecutar un piloto de utilización de la metodología.

Se formuló la siguiente *Hipótesis de investigación*:

El diseño y aplicación de una metodología de evaluación de arquitectura de software en los proyectos productivos de la UCI contribuirá a mejorar la calidad del proceso de desarrollo y del producto final.

Para lograr los objetivos trazados y demostrar la hipótesis, se acometieron las siguientes *Tareas de investigación*:

- ✓ Analizar la evolución del enfoque arquitectónico en el desarrollo de software.
- ✓ Estudiar las definiciones de arquitectura de software dadas por diferentes autores.
- ✓ Analizar modelos de calidad que guíen la selección de las características de calidad, los atributos y las métricas para evaluar la arquitectura de software como producto intermedio.
- ✓ Caracterizar las técnicas de evaluación de arquitectura de software.
- ✓ Caracterizar los métodos de evaluación de arquitectura de software.
- ✓ Comparar las técnicas y los métodos existentes teniendo en cuenta el enfoque de una evaluación arquitectónica integral.
- ✓ Estudiar el proceso de la actividad productiva en la UCI.
- ✓ Realizar encuestas y entrevistas al personal involucrado en la actividad productiva en la UCI.
- ✓ Realizar encuestas y entrevistas a personas especializadas en el proceso de aseguramiento de la calidad en la producción de software, la evaluación arquitectónica y el cumplimiento de la calidad de los productos de software.
- ✓ Diseñar una metodología de evaluación de arquitectura teniendo en cuenta, actividades, involucrados y documentación que se genera.

- ✓ Definir las distintas fases que debe abarcar la metodología de evaluación de arquitectura de software.
- ✓ Definir las actividades de cada una de las fases.
- ✓ Definir los involucrados de las actividades, así como los artefactos de entrada y salida.
- ✓ Definir un conjunto de buenas prácticas en cada de una de las actividades, basándose en la experiencia de proyectos internacionales.
- ✓ Definir que modelo de calidad usaría la metodología que permita la selección y verifique el cumplimiento de los atributos de calidad en la arquitectura del sistema.
- ✓ Identificar un método para la evaluación técnica del modelo.
- ✓ Elaborar los elementos necesarios para aplicar el método de evaluación técnica.
- ✓ Definir un proyecto piloto para utilizar la metodología.
- ✓ Evaluar la arquitectura diseñada por el proyecto a través de la metodología propuesta.
- ✓ Evaluar los resultados de la aplicación de la metodología de evaluación de arquitectura en el proyecto piloto.

Diseño metodológico de la investigación.

La *población* a estudiar en el presente trabajo será:

- ✓ Todo el personal vinculado al proceso de definición y evaluación de las arquitecturas de software en los proyectos productivos de la UCI. Incluye líderes de proyecto, arquitectos y asesores de calidad.

Unidad de estudio:

- ✓ Personal vinculado al proceso de definición y evaluación de la arquitectura de software.

Tamaño de la muestra:

Con el objetivo de garantizar que todas las características de la población estén representadas en la muestra, y teniendo en cuenta el alto número de proyectos

productivos de la UCI, se utilizó el Muestreo Estratificado. Las subpoblaciones en que fue dividida la población heterogénea fueron las siguientes:

- ✓ Líderes de proyecto: Son las personas que dirigen los proyectos de desarrollo de software en la universidad.
- ✓ Arquitectos: Son las personas que definen la arquitectura, establecen las tecnologías a usar y las guías para la codificación, entre otras actividades.
- ✓ Asesores de calidad: Es el personal que diseña y controla el aseguramiento de la calidad de un producto durante su ciclo de desarrollo, garantizando el cumplimiento de la calidad interna y externa del mismo.

Para elegir la muestra de las distintas subpoblaciones se realizó un muestreo aleatorio simple. La muestra está formada de acuerdo a la subpoblación de la siguiente manera:

- ✓ Líderes de proyecto: 17, que constituye el 12 % de la subpoblación.
- ✓ Arquitectos: 20, que constituye el 15 % de la subpoblación.
- ✓ Asesores de calidad: 14, que constituye el 10 % de la subpoblación.

Métodos utilizados en la investigación.

Métodos Teóricos

- ✓ Histórico-lógico. Se utilizó para estudiar el objeto y campo de acción de la investigación a través del tiempo y determinar las tendencias existentes en la evaluación de arquitecturas de software.
- ✓ Modelación: Posibilitó la modelación de los componentes de la metodología de evaluación que se propondrá.
- ✓ Análisis y síntesis: Se utilizaron en el estudio y valoración de las técnicas y métodos de evaluación de arquitectura de software existente, así como los modelos de calidad analizados, que servirán de fundamento teórico de la investigación.

Métodos empíricos.

- ✓ Revisión de documentos: Se revisaron documentos emitidos por el Instituto de Ingeniería de Software (Software Engineering Institute, SEI) y el Grupo de Revisión y Evaluación de Arquitectura de Software (Software

Architecture Review and Assessment, SARA) de la Universidad Técnica de Eindhoven; donde se exponen casos de estudio, que utilizan métodos de evaluación de arquitecturas, brinda resultados y dificultades de cada uno, de forma valorativa y comparativa. Unido a esto, se consultaron las normas ISO 9126-1, 2 y 3, ISO 14598.

- ✓ Observación: Este método permitió obtener la información del comportamiento del objeto de investigación tal y como éste se da en la realidad, es decir, fue una forma de obtener información directa e inmediata sobre el fenómeno.
- ✓ Encuestas: Se utilizaron para obtener los problemas presentes en el proceso de evaluación de arquitectura de software, criterios acerca de la metodología propuesta y para evaluar los resultados.
- ✓ Entrevistas: Este método se utilizó para validar la propuesta, al entrevistar al personal especializado en temáticas como el aseguramiento de la calidad en los proyectos y arquitectura de software.

Estructura de este documento.

Capítulo 1: Este capítulo enuncia la evolución que ha tenido el concepto de arquitectura de software desde sus inicios. Elementos de por qué es importante la definición de la arquitectura, y como la evaluación de la misma impacta de manera directa sobre la calidad del proceso de desarrollo y del producto final. Se analizan los distintos modelos enfocados a la calidad del producto, con el objetivo de seleccionar el más adecuado en la verificación de los requerimientos de calidad que debe cumplir la arquitectura de software, como producto intermedio del ciclo de desarrollo. Se exponen las técnicas y métodos de evaluación de arquitecturas existentes, y una panorámica de las principales entidades internacionales, que han investigado y puesto en práctica estas premisas con el objetivo de evaluar arquitecturas de software.

Capítulo 2: En este capítulo se realiza un proceso de caracterización de la actividad productiva de la UCI, el cual sirve de basamento para el diseño de la metodología de evaluación de arquitectura. Se analizan los métodos científicos utilizados en la investigación y con que finalidad se usaron cada uno de ellos. Se fundamenta el método de evaluación técnica que valora la opinión de los expertos

en la aceptación de la propuesta o no. Finalmente se realiza la propuesta de la metodología de evaluación arquitectónica con las fases y actividades que la conforman, los involucrados y los artefactos que genera.

Capítulo 3: En este capítulo describe los pasos a seguir para lograr una aplicación exitosa del uso de la metodología de evaluación de arquitectura de software en los proyectos productivos de la universidad. Realiza un análisis de los resultados y las encuestas aplicadas, con el objetivo de establecer un nivel de partida que contribuya a la comparación con los resultados de la aplicación. Analiza los resultados de la evaluación técnica de la propuesta brindando una panorámica de la opinión de los expertos y finalmente realiza un análisis valorativo de los resultados del uso de la metodología en un proyecto piloto.

Conclusiones y Recomendaciones: Se plantean si fueron cumplidos los objetivos de la presente investigación y los aportes prácticos de la misma. Se realizan un conjunto de recomendaciones con el objetivo de perfeccionar y ampliar el alcance de la investigación.

Bibliografía: Se reflejan todas las fuentes bibliográficas consultadas y referenciadas para la realización de la investigación.

Anexos: Se muestran tablas, plantillas, modelos de encuestas y entrevistas que apoyaron la investigación durante su desarrollo.

Capítulo 1. Introducción a la evaluación de las arquitecturas de software

1.1 Introducción.

Este capítulo enuncia la evolución que ha tenido el concepto de arquitectura de software desde sus inicios. Elementos de por qué es importante la definición de la arquitectura, y como la evaluación de la misma impacta de manera directa sobre la calidad del proceso de desarrollo y del producto final. Se analizan los distintos modelos enfocados a la calidad del producto, con el objetivo de seleccionar el más adecuado en la verificación de los requerimientos de calidad que debe cumplir la arquitectura de software, como producto intermedio del ciclo de desarrollo. Se exponen las técnicas y métodos de evaluación de arquitecturas existentes, y una panorámica de las principales entidades internacionales, que han investigado y puesto en práctica estas premisas con el objetivo de evaluar arquitecturas de software.

1.2 La arquitectura de software: evolución y definición de un concepto.

Los primeros intentos por definir a la arquitectura de software fueron los expuestos por Mary Shaw en el año 1989, en su libro “*Larger Scale Systems Require Higher-Level Abstractions*”, presentado en el 5to Seminario Internacional sobre la especificación y el diseño del software, y publicado por la IEEE Computer Society. Shaw define la arquitectura de la siguiente manera (Shaw, 1989):

...“Arquitectura de software es el estudio de la estructura a gran escala y el rendimiento de los sistemas de software. Aspectos importantes de la arquitectura de un sistema incluye la división de funciones entre los módulos del sistema, los medios de comunicación entre módulos, y la representación de la información compartida.”

Se observa como la concepción de componentes, “división de funciones entre los módulos del sistema”, y de conectores, “medios de comunicación entre módulos”, van tomando su espacio dentro de la definición.

Ya en 1990, Thomas G. Lane, logra publicar en el SEI, el libro “*Studying Software Architecture Through Design Spaces and Rules*”, que constituye el primer ejemplar publicado por el Instituto acerca del tema de la arquitectura de software. En este libro, Lane brinda un concepto interesante, *espacios de diseño*, los cuales proporcionan un marco para la confección de normas que pueden ayudar a un diseñador en la selección de una arquitectura apropiada para las necesidades funcionales de un nuevo sistema. Resulta interesante, debido a que se observa como el rol del arquitecto de software, aún no es reconocido, esta tarea es asignada directamente al diseñador del software (Lane, 1990).

Luego de este aporte, se observa una tendencia marcada hacia la búsqueda de un modelo de estructuración de software, y se diseñan un conjunto de modelos de dominios, basados en diseños genéricos, como por ejemplo DSSA (*Domain-Specific Software Architectures*) elaborado por Mettala y Graham, en el año 1992. (Mettala, y otros, 1992)

No es hasta el lanzamiento del libro “*An Introduction to Software Architecture*”, de Mary Shaw y David Garlan, en el año 1994, donde plantean, que debido al aumento del tamaño y complejidad de los productos de software, el problema principal no radica ya, en los algoritmos y las estructuras de datos, sino que se dirige a la organización de los componentes que conforman el sistema, introduciendo así, la necesidad de la creación de la arquitectura de software, como disciplina científica, cuyo objeto de estudio no es más que la determinación de un conjunto de paradigmas que establezcan una organización del sistema a alto nivel, la interrelación entre los distintos componentes que lo conforman y los principios que orientan su diseño y evolución. (Garlan, y otros, 1994)

En esta ponencia se introduce por primera vez el término estilos arquitectónicos, y se definen una gran gama de ellos, entre los que se encuentran: Tubería y filtros, Repositorio, Arquitectura en Capas, Arquitectura basada en eventos, etc.

En este mismo año, 1994, el cual podría ser considerado como el año de oro de la arquitectura de software, se realiza otro acontecimiento arquitectónico relevante. Mary Shaw y David Garlan, lanzan el concepto de Lenguajes de Descripción Arquitectónica (ADLs), en su libro “*Characteristics of Higher Level Languages for Software Architecture*”. En el estudio se señala las alternativas que hasta el momento se poseen para la definición de la arquitectura de software de un sistema.

En primer lugar, a través de la modularización de las herramientas existentes de programación y de los módulos de conexión entre ellas y, en segundo lugar, describir sus diseños usando diagramas informales y frases idiomáticas. A partir de estas reflexiones, definen un conjunto de regularidades y propiedades específicas, que constituyeron las bases de los ADLs (Shaw, y otros, 1994).

Actualmente en la literatura (Bass y otros, 1998; Kazman y otros, 2001; Hofmeister y otros, 2000; Lane, 1990; Buschman y otros, 1996; Booch y otros, 1999; Abowd, 1995), es posible encontrar numerosas definiciones del término arquitectura de software, cada una con planteamientos diversos. Se hace evidente que su conceptualización sigue todavía en discusión, puesto que no es posible referirse a un diccionario en busca de un significado, y tampoco existe un estándar que pueda ser tomado como marco de referencia.

Sin embargo, al hacer un análisis detallado de cada uno de los conceptos disponibles, resulta interesante la existencia de ideas comunes entre los mismos, sin observarse planteamientos contradictorios, sino más bien complementarios.

De aquí que la mayoría de los autores (Bass y otros, 1998; Kazman y otros, 1998; Hofmeister y otros, 1995; Lane, 1990; Buschman y otros, 1996; Booch y otros, 1999; Abowd, 1995) coinciden en que una arquitectura de software define la estructura del sistema. Esta estructura se constituye de componentes, módulos o piezas de código que nacen de la noción de abstracción, cumpliendo funciones específicas, e interactuando entre sí con un comportamiento definido (Bass y otros, 1998; Hayes-Roth, 1995; Hofmeister y otros, 2000; Buschman y otros, 1996; Booch y otros, 1999; Abowd, 95). Los componentes se organizan de acuerdo a ciertos criterios, que representan decisiones de diseño. En este sentido, hay autores que plantean que la arquitectura de software incluye justificaciones referentes a la organización y el tipo de componentes, garantizando que la configuración resultante satisface los requerimientos del sistema (Boehm y otros, 1995).

En pocas palabras se puede definir a la arquitectura de software como la estructura del sistema en función de la definición de los componentes y sus interacciones. (Bass, y otros, 2005) Esta es la definición de arquitectura de software en la que se basará esta investigación para su estudio, debido a la precisión y sencillez del concepto.

La arquitectura de software ocupa un lugar importante en el proceso de desarrollo del software debido a diferentes aspectos como:

- ✓ *Permite la comunicación entre las partes interesadas.* La arquitectura de software representa un resumen de la descripción del sistema a un nivel de abstracción suficientemente alto, que permite la comunicación entre los diferentes actores del sistema.
- ✓ *Permite, luego de su definición, mayor claridad y especificación en el proceso de desarrollo.* Al quedar establecida una arquitectura de software, luego de realizar un análisis valorativo del cumplimiento de los requerimientos de calidad internos, tanto los desarrolladores como demás integrantes del equipo cuentan con una línea base diseñada que les brinda mayor claridad del producto a desarrollar, las interacciones que se establecen entre los distintos componentes del sistema, los elementos reutilizables, etc. Todos estos aspectos contribuyen en gran medida a aumentar y mejorar la calidad del proceso de desarrollo, a reducir tiempo, y a su vez costos, e impacta favorablemente en la calidad del producto final.
- ✓ *Permite la reutilización mediante la consolidación del manejo de las abstracciones de un sistema.* La arquitectura de software representa un conciso plan de diseño de un sistema. Este plan de diseño se puede aplicar a otros sistemas que presentan necesidades similares, como por ejemplo, una misma línea de productos.

1.3 Calidad arquitectónica.

El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades.

A la hora de definir la calidad del software se pueden adoptar diferentes aproximaciones. Como primera aproximación es importante diferenciar entre la calidad del *producto* software y la calidad del *proceso* de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

Se plantea que la calidad de software "...es la suma de todos aquellos aspectos o características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades, expresadas o implícitas." (ISO-8402, 1994)

Otras definiciones son:

- ✓ "...la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente." (Pressman, 1998)
- ✓ "Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas" (IEEE-729, 1983)
- ✓ Capacidad del producto software para satisfacer los requisitos establecidos. (DoD-STD-2168, 1985)

En la definición de la calidad del software pueden estar involucrados aspectos como la ausencia de defectos, aptitud para el uso, seguridad, confiabilidad, entre otras cualidades. Sin embargo, hay algo importante que se debe tener presente: la calidad del software debe ser construida desde el comienzo, no es algo que puede ser añadido después.

Evaluar un producto de software implica conocer aspectos de su calidad. La funcionalidad, eficiencia y capacidad de mantenimiento son características de calidad del producto. La arquitectura de software constituye un producto intermedio que se genera durante el ciclo de desarrollo de un producto, por lo que puede ser evaluada a través del cumplimiento de los requerimientos exigidos por modelos y estándares internacionales que miden la calidad del producto final.

La calidad del diseño arquitectónico, como se ha plateado anteriormente, es de suma importancia durante el proceso de desarrollo de software, pues contribuye en alto grado al aumento de la calidad del producto final.

Con el objetivo de analizar, certificar y mejorar la calidad de la arquitectura como producto, es necesario tener como base un modelo de calidad que guíe la evaluación. Con la premisa de lograr este objetivo fueron estudiados distintos modelos de calidad: el modelo de McCall, Boehm, Dromey, FURPS, ISO 9126-1, buscando identificar cual de ellos realiza un enfoque más cercano a la evaluación de la calidad arquitectónica.

1.3.1 Modelos de calidad.

Los modelos de calidad del software vienen a ayudar en la puesta en práctica del concepto general de calidad, ofreciendo una definición más operacional. En los modelos, la calidad se define de forma jerárquica. Es un concepto que se deriva en un conjunto de sub-conceptos, cada uno los cuales se van a evaluar a través de un conjunto de indicadores o métricas. Tienen una estructura, por lo general, en tres niveles:

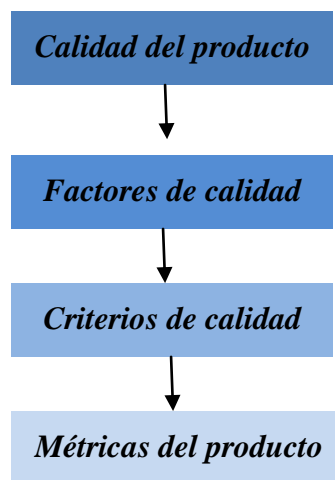


Fig. 1 Descomposición jerárquica de los modelos de calidad

- ✓ En el nivel más alto de la jerarquía se encuentran los *factores* de calidad, que representan la calidad desde el punto de vista del usuario. Son las características que componen la calidad. También se les conoce como atributos de calidad externos.
- ✓ Cada uno de los factores se descompone en un conjunto de *criterios* de calidad. Estos son atributos que, cuando están presentes, contribuyen al aspecto de la calidad que el factor asociado representa. Se trata de una visión de la calidad desde el punto de vista del equipo de desarrollo. También se les conoce como los atributos de calidad internos.
- ✓ Para cada uno de los criterios de calidad se definen un conjunto de métricas, que son medidas cuantitativas de ciertas características del producto que, cuando están presentes, dan una indicación del grado en que dicho producto posee un determinado atributo de calidad.

La ventaja de los modelos de calidad es que esta se convierte en algo concreto, que se puede definir, que se puede medir y, sobre todo, que se puede planificar. Los modelos de calidad contribuyen además a comprender las relaciones que existen entre diferentes características de un producto software.

1.3.1.1 Modelo de McCall.

El modelo, propuesto por McCall en 1977 (Gillies, 1997) está orientado a los desarrolladores de sistemas, para ser utilizado durante el proceso de desarrollo. Este abarca tres áreas de trabajo:

Área de trabajo	Descripción
Operación del producto	Requiere que pueda ser comprendida rápidamente, operada eficientemente y que los resultados sean aquellos requeridos por el usuario
Revisión del producto	Está relacionada con la corrección de errores y la adaptación de los sistemas. Esto es importante porque es generalmente considerada como la parte más costosa en el desarrollo de software.
Transición del producto	Puede que no sea muy importante en todas las aplicaciones. Sin embargo, la orientación a procesamiento distribuido y el rápido cambio en el hardware es probable que incremente su importancia.

Tabla 1. Áreas de trabajo direccionadas por el modelo de McCall. (Gillies, 1997)

Las tres áreas que contempla el Modelo de McCall abarcan los siguientes factores: (Pressman, 1998)

- Operación del producto:
 - ✓ **Corrección.** Hasta dónde satisface un programa su especificación y logra los objetivos de la misión del cliente.
 - ✓ **Fiabilidad.** Hasta dónde se espera que un programa lleve a cabo su función pretendida con la exactitud requerida.
 - ✓ **Eficiencia.** Cantidad de recursos informáticos y código necesarios para que un programa realice su función.
 - ✓ **Integridad.** Hasta dónde se controla el acceso al software o a los datos por personas no autorizadas.

- ✓ **Usabilidad** (Facilidad de manejo). Esfuerzo necesario para aprender, operar, preparar los datos de entrada e interpretar las salidas (resultados) de un programa.
- Revisión del producto:
 - ✓ **Mantenibilidad** (Facilidad de mantenimiento). Esfuerzo necesario para localizar y arreglar un error del programa.
 - ✓ **Flexibilidad**. Esfuerzo necesario para modificar un programa operativo.
 - ✓ **Facilidad de prueba**. Esfuerzo necesario en probar un programa para asegurarse que realiza su función pretendida.
- Transición del producto:
 - ✓ **Portabilidad**. Esfuerzo necesario para transferir el programa de un entorno de sistema de hardware y/o de software a otro.
 - ✓ **Reusabilidad** (capacidad de reutilización). Hasta dónde se vuelve a emplear un programa [o partes de un programa] en otras aplicaciones, en relación al empaquetamiento y alcance de las funciones que realiza el programa.
 - ✓ **Interoperabilidad**. Esfuerzo necesario para acoplar un sistema con otro.

A cada uno de estos factores se asocian un conjunto de criterios, que se observan en la siguiente tabla:

Factor	Criterio
Corrección	<ul style="list-style-type: none"> ✓ Trazabilidad ✓ Completitud ✓ Consistencia
Fiabilidad	<ul style="list-style-type: none"> ✓ Consistencia ✓ Exactitud ✓ Tolerancia a fallos
Eficiencia	<ul style="list-style-type: none"> ✓ Eficiencia de ejecución ✓ Eficiencia de almacenamiento
Integridad	<ul style="list-style-type: none"> ✓ Control de acceso ✓ Auditoría de acceso
Usabilidad	<ul style="list-style-type: none"> ✓ Operatividad ✓ Entrenamiento ✓ Comunicación
Mantenibilidad	<ul style="list-style-type: none"> ✓ Simplicidad ✓ Concreción
Flexibilidad	<ul style="list-style-type: none"> ✓ Auto-descripción ✓ Capacidad de expansión ✓ Generalidad ✓ Modularidad

Facilidad de prueba	<ul style="list-style-type: none"> ✓ Simplicidad ✓ Instrumentación ✓ Auto-descripción ✓ Modularidad
Portabilidad	<ul style="list-style-type: none"> ✓ Auto-descripción ✓ Independencia del sistema ✓ Independencia del hardware
Reusabilidad	<ul style="list-style-type: none"> ✓ Auto-descripción ✓ Generalidad ✓ Modularidad ✓ Independencia del sistema ✓ Independencia del hardware
Interoperabilidad	<ul style="list-style-type: none"> ✓ Modularidad ✓ Similitud de comunicación ✓ Similitud de datos

Tabla 2. Criterios asociados a los factores de calidad (McCall, y otros, 1977)

En este modelo, el término *factor de calidad* define características claves que un producto debe exhibir. Los atributos del factor de calidad que define el producto son los nombrados *criterios de calidad*. Las *métricas de calidad* denotan una medida que puede ser utilizada para cuantificar los criterios. Las métricas desarrolladas están relacionadas con los factores de calidad y la relación que se establece se mide en función del grado de cumplimiento de los criterios.

El modelo propone 41 métricas, sobre todo métricas de tipo subjetivo, es decir, métricas que evaluadas por personas diferentes podrían dar como resultado valores diferentes. En general, el modelo de McCall asocia a cada criterio una fórmula de regresión del tipo:

$$CC = r_1 m_1 + r_2 m_2 + \dots + r_n m_n$$

donde los **r_j** son los coeficientes de regresión, y donde los **m_j** representan las distintas métricas asociadas al criterio. De la misma forma se propagarán a los factores los valores calculados para los criterios. Por ejemplo

La medida para el factor corrección, por ejemplo, se calculará aplicando la fórmula:

$$\frac{x+y+z}{3}$$

donde x es la medida para la completitud, y la medida para la trazabilidad y z la medida para la consistencia. Para que todas las métricas se puedan combinar sin problemas, lo habitual es que se normalicen sus valores dentro del intervalo [0,1].

El modelo de McCall constituye uno de los modelos más difundidos y ha servido de base para el desarrollo de nuevos modelos. Este modelo emplea la descomposición jerárquica en árbol para la medida de la calidad y tiene como principal desventaja que es fijo, sin posibilidades de ser modificado.

1.3.1.2 Modelo de Boehm.

Unos de los modelos que tomó como base el modelo de McCall fue el modelo de Boehm (Berander, y otros, 2005), otro predecesor de los modelos de calidad actuales, creado por Barry W. Boehm en el año 1978. Este modelo fue el primero en proponer la evaluación de la calidad del software por medio de métodos automáticos y cuantitativos.

Este modelo incluye las necesidades de los usuarios, como lo hace Mc Call; sin embargo, incluye características de rendimiento de hardware que no se encuentran en su predecesor. Boehm, al igual que McCall, es un caso clásico del uso de la descomposición arbórea para las medidas de calidad, a continuación se describe su árbol de caracterización de la calidad del producto:

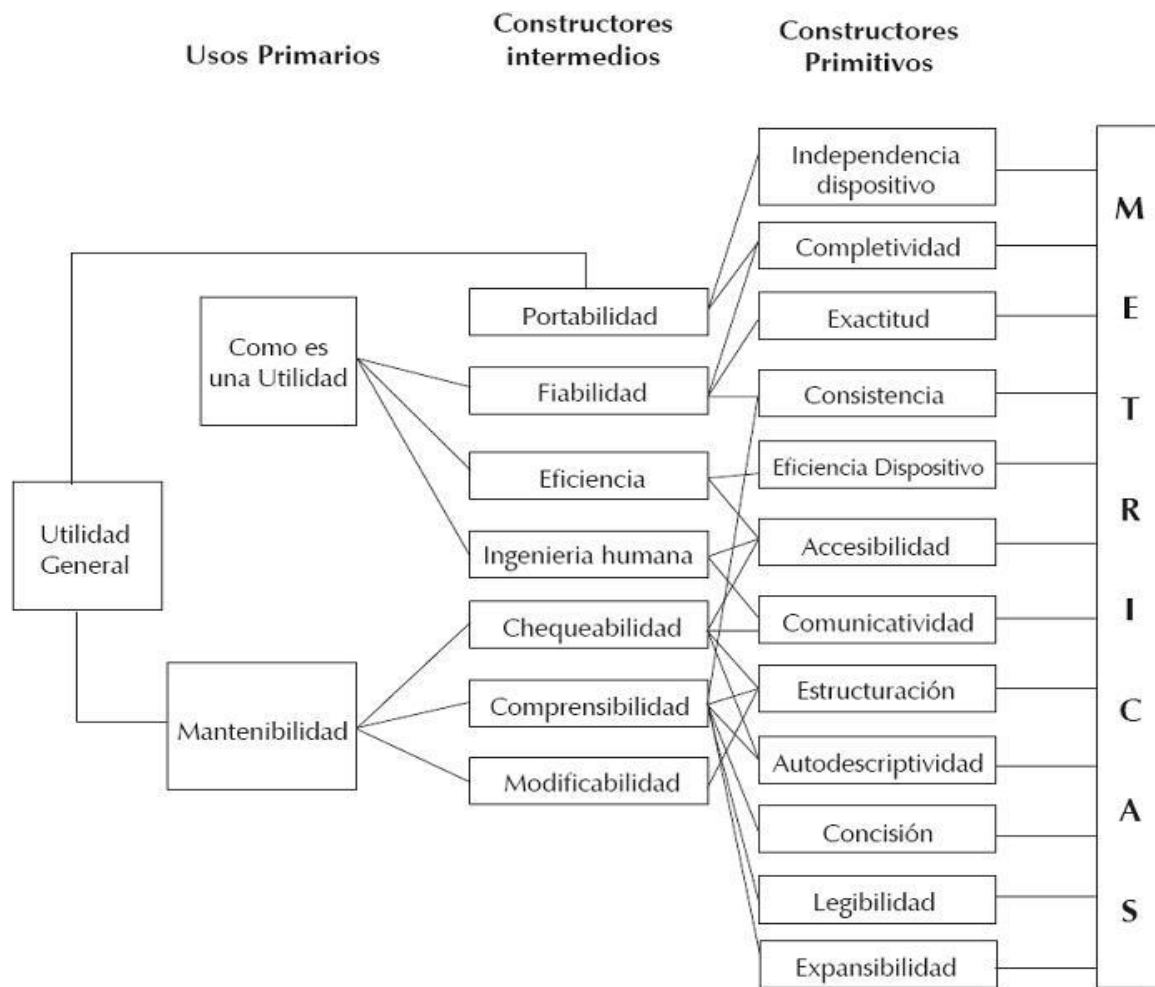


Fig. 2. Árbol de calidad del Modelo de Boehm.

Una conclusión importante que aporta este modelo es que no es posible resumir en un solo número todos los componentes y métricas de calidad, sobre todo por que algunos componentes son conflictivos entre sí. Por esto de acuerdo a las necesidades que se tienen del software, es que se derivan un orden de prioridades que le permite al desarrollador tomar decisiones sobre qué tipo de atributos representan en cada caso la calidad del producto.

Al igual que el modelo de McCall, en el modelo de Boehm los factores se encuentran interrelacionados y el usuario no puede modificarlos o adaptarlos, solamente usarlos tal como se presentan.

Además de que se observan similitudes entre ambos modelos, la diferencia está en que McCall focaliza en medidas precisas de alto nivel, mientras que Boehm presenta un rango más amplio de características primarias. La mantenibilidad está más desarrollada en Boehm. A continuación se expone una tabla comparativa de ambos modelos:

Criterio	McCall	Boehm	Criterio	McCall	Boehm
correctitud	+	+	confiabilidad	+	+
integridad	+	+	usabilidad	+	+
eficiencia	+	+	mantenim.	+	+
testeabilidad	+		interoperab.	+	
flexibilidad	+	+	reusabilidad	+	+
portabilidad	+	+	claridad		+
modificab.		+	document.		+
entendib.		+	validez		+

Fig. 3 Comparación entre el modelo de McCall y Boehm.

1.3.1.3 Modelo de Dromey

Dromey (Dromey, 1996) propuso un marco de referencia – o metamodelo - para la construcción de modelos de calidad, basado en cómo las propiedades medibles de un producto de software pueden afectar los atributos de calidad generales, como por ejemplo, confiabilidad y mantenibilidad. El problema que se plantea es cómo conectar las propiedades del producto con los atributos de calidad de alto nivel.

Para solventar esta situación, Dromey (Dromey, 1996) sugiere el uso de cuatro categorías que implican propiedades de calidad, que son: corrección, internas, contextuales y descriptivas, las cuales se describen a continuación:

1. **Corrección.** Pueden ser internas (asociadas con los componentes individuales) o contextuales (asociadas con la manera en que los componentes son utilizados en el contexto)
2. **Internas.** Miden que tan bien un componente ha sido entregado de acuerdo a su objetivo, implementación o que tan bien ha sido compuesto.
3. **Contextuales.** Cómo los componentes son compuestos y las influencias que ejercen sobre la calidad del producto.
4. **Descriptivas.** Para ser útil un software debe ser fácil de entender y utilizar de acuerdo a su propósito. Estas propiedades descriptivas aplican a requerimientos, diseños, implementación y a las interfaces de usuario.

.La tabla 4 presenta la relación que establece Dromey (Dromey, 1996) entre las propiedades de calidad del producto y los atributos de calidad de alto nivel.

Propiedades del producto	Atributos de calidad
Corrección	<ul style="list-style-type: none"> ✓ Funcionalidad ✓ Confiabilidad
Internas	<ul style="list-style-type: none"> ✓ Mantenibilidad ✓ Eficiencia ✓ Confiabilidad
Contextuales	<ul style="list-style-type: none"> ✓ Mantenibilidad ✓ Reusabilidad ✓ Portabilidad ✓ Confiabilidad
Descriptivas	<ul style="list-style-type: none"> ✓ Mantenibilidad ✓ Reusabilidad ✓ Portabilidad ✓ Usabilidad

Tabla 3. Relación entre propiedades del producto y atributos de calidad. (Dromey, 1996)

El proceso de construcción de modelos de calidad propuesto por Dromey (Dromey, 1996) consta de 5 pasos, basados en las propiedades mencionadas. Los pasos del marco de referencia propuesto son:

1. Especificación de los atributos de calidad de alto nivel (por ejemplo, confiabilidad, mantenibilidad).
2. Determinación de los distintos componentes del producto a un apropiado nivel de detalle (por ejemplo, paquetes, subrutinas, declaraciones).
3. Para cada componente, determinación y categorización de sus implicaciones más importantes de calidad.
4. Proposición de enlaces que relacionan las propiedades implícitas a los atributos de calidad, o, alternativamente, uso de enlaces de las cuatro categorías de atributos propuestas.
5. Iteración sobre los pasos anteriores, utilizando un proceso de evaluación y refinamiento.

Para ilustrar sus planteamientos, Dromey (Dromey, 1996) demuestra el uso de su procedimiento para la construcción de un modelo de calidad de implementación, un modelo de calidad de requerimientos, y un modelo de calidad de diseño.

1.3.1.4 Modelo *FURPS*

Este modelo fue presentado por Hewlett-Packard en el año 1987. En él se desarrollan un conjunto de factores de calidad de software, bajo el acrónimo de *FURPS*: funcionalidad (Functionality), usabilidad (Usability), confiabilidad (Reliability), desempeño (Performance) y capacidad de soporte (Supportability). (Grady, y otros, 1987)

La tabla 5 presenta la clasificación de los atributos de calidad que se incluyen en el modelo, junto con las características asociadas a cada uno (Pressman, 2002):

Factor de calidad	Atributos
Funcionalidad	<ul style="list-style-type: none">✓ Características y capacidades del programa✓ Generalidad de las funciones✓ Seguridad del sistema
Facilidad de uso	<ul style="list-style-type: none">✓ Factores humanos✓ Factores estéticos✓ Consistencia de la interfaz✓ Documentación
Confiabilidad	<ul style="list-style-type: none">✓ Frecuencia y severidad de las fallas✓ Exactitud de las salidas✓ Tiempo medio de fallos✓ Capacidad de recuperación ante fallas✓ Capacidad de predicción
Rendimiento	<ul style="list-style-type: none">✓ Velocidad del procesamiento✓ Tiempo de respuesta✓ Consumo de recursos✓ Rendimiento efectivo total✓ Eficacia
Capacidad de soporte	<ul style="list-style-type: none">✓ Extensibilidad✓ Adaptabilidad✓ Capacidad de pruebas✓ Capacidad de configuración✓ Compatibilidad✓ Requisitos de instalación

Tabla 4. Atributos de calidad. (Grady, y otros, 1987)

Los factores de calidad FURPS y sus atributos, pueden usarse para establecer métricas de la calidad para todas las actividades del proceso de desarrollo de sistemas de información.

El modelo FURPS incluye, además de los factores de calidad y los atributos, restricciones de diseño y requerimientos de implementación, físicos y de interfaz (Grady, y otros, 1987). Las restricciones de diseño especifican o restringen el diseño del sistema. Los requerimientos de implementación especifican o restringen la codificación o construcción de un sistema (por ejemplo, estándares requeridos, lenguajes, políticas). Por su parte, los requerimientos de interfaz especifican el comportamiento de los elementos externos con los que el sistema debe interactuar. Por último, los requerimientos físicos especifican ciertas propiedades que el sistema debe poseer, en términos de materiales, forma, peso, tamaño (por ejemplo, requisitos de hardware, configuración de red).

1.3.1.5 Modelo ISO/IEC 9126

La Organización Internacional para la Estandarización (ISO, *International Organization for Standardization*) ha emitido algunas normas que definen un modelo de calidad del software, en varios contextos de uso:

- ✓ ISO 9126-1 - define 6 características de calidad principales, y 27 subcaracterísticas. Incluye 3 reportes técnicos (ISO/IEC 9126-2, 3 e 4)
- ✓ ISO/IEC 9241 - define las características de un software *usable*.
- ✓ ISO 12119 - define las características de calidad para un software COTS (Commercial off the shelf)
- ✓ Ha publicado además la norma 14598 que guía en el proceso de valoración de la calidad del software según los criterios de la 9126.

Durante muchos años se buscó en la Ingeniería de Software un modelo único para expresar calidad, la ventaja era obvia: poder comparar productos entre sí. En 1992, una variante del modelo de McCall fue propuesta como estándar internacional para la medición de calidad de software. Así surge el estándar ISO/IEC 9126, formalmente nombrado Software Product Evaluation: Quality Characteristics and Guidelines for their Use, el cual ha sido desarrollado en un

intento de identificar los atributos clave de calidad para un producto de software (Pressman, 2002).

Este modelo identifica seis factores de calidad que pueden estar presentes en cualquier producto de software, y pueden ser considerados tanto internos como externos:

- ✓ Funcionalidad
- ✓ Confiabilidad
- ✓ Eficiencia
- ✓ Usabilidad
- ✓ Mantenibilidad
- ✓ Portabilidad

Además de estos factores esta norma define cuatro factores de calidad en uso:

- **Eficacia.** Capacidad de ayudar al usuario a realizar sus objetivos con exactitud y completitud, en un contexto acordado.
- **Productividad.** Capacidad de ayudar al usuario en emplear una apropiada cantidad de recursos en obtener sus resultados.
- **Satisfacción.** Capacidad de satisfacer a un usuario en un contexto de uso determinado.
- **Seguridad.** Capacidad de lograr aceptables niveles de riesgo para las personas, el ambiente de trabajo, y la actividad, en un contexto de uso establecido.

A través de este modelo se pretende que estas características mencionadas sean comprensivas, es decir, cualquier componente de la calidad de software se puede describir como combinación de aspectos de estos factores. A su vez estas características incluyen un conjunto de subcaracterísticas que pueden ser refinadas mediante múltiples niveles.

Es interesante destacar que los factores de calidad que contempla el estándar ISO/IEC 9126 no son necesariamente usados para mediciones directas (Pressman, 2002), pero proveen una valiosa base para medidas indirectas, y una excelente lista para determinar la calidad de un sistema.

Para este modelo de calidad se propone una adaptación para efectos específicos de la evaluación de la arquitectura de software como producto intermedio del ciclo de desarrollo. (Losavio, y otros, 2003)

El modelo se basa en los atributos de calidad que se relacionan directamente con la arquitectura: funcionalidad, confiabilidad, eficiencia, mantenibilidad y portabilidad. Los autores plantean que la característica usabilidad propuesta por el modelo ISO/IEC 9126 puede ser refinada para obtener atributos que se relacionan con los componentes de la interfaz con el usuario. Dado que estos componentes son independientes de la arquitectura, no son considerados en la adaptación del modelo. (Losavio, y otros, 2003)

Una arquitectura diseñada, que exprese a un alto nivel todos los componentes que conforman el sistema, los conectores y las relación entre ellos, posea un alto grado de configuración o topología, debe satisfacer las seis características que establece el modelo de calidad ISO 9126, o un subconjunto de ellas.

A continuación se exponen cómo los requisitos de calidad son refinados y cómo se adaptan a la arquitectura de software, teniendo en cuenta que las características y sub-características se consideran independientes.

Funcionalidad.

Adecuación. Esta sub-característica se basa en que el producto posea las funciones necesarias para las tareas que debe desempeñar. Esto se puede observar a través de dos aspectos, las tareas que se especifican a través de los casos de uso, deben tener asociadas una funcionalidad que permita materializarlas, y el refinamiento de cada una de estas tareas debe ser especificado mediante los diagramas de secuencia.

A nivel arquitectónico:

En este caso se identifican todas las funcionalidades del sistema y cada una de ellas se refina en un atributo cuyo valor es sí (1) o no (0). Observar que pueden definirse atributos que tengan como rango de valor [1...0], expresando sentido de presencia o ausencia del atributo. La métrica en este caso constituye una escala para obtener niveles de calificación.

La secuencia de los diagramas obtenidos a partir de los requisitos funcionales se refina. En caso de haber una especificación de la arquitectura, la funcionalidad

especificada se descompone en funciones asociadas a los componentes y esta composición se verifica contra los requerimientos del sistema.

Exactitud. Esta sub-característica proporciona que los resultados sean los correctos o los esperados con un grado necesario de precisión. Puede ser medida por un atributo que verifique a través del código fuente, o a través de la medición de las funciones de cada componente.

A nivel arquitectónico:

- ✓ Se identifica los componentes que son responsables de funciones vinculadas directamente a los cálculos de precisión.
- ✓ El atributo se calcula mediante las siguientes mediciones:

Π_i (Exactitud (componentes funcionales i))

Interoperabilidad. Esta sub-característica evidencia la capacidad del producto de establecer una interacción entre sus componentes y/o con otros sistemas externos. Los autores plantean que se utiliza en lugar de compatibilidad para evitar la ambigüedad con la sustitución.

A nivel arquitectónico:

- ✓ Identificar los conectores de comunicación con sistemas externos. Por ejemplo, para exigir la compatibilidad CORBA implica la existencia de componentes CORBA.
- ✓ Es refinada en un atributo cuyo valor es sí o no.

Seguridad. La capacidad para prevenir el acceso no autorizado a programas o datos.

A nivel arquitectónico:

- ✓ Significa contar con un mecanismo o dispositivo (software o hardware) para llevar a cabo esta tarea de forma explícita. Puede ser un componente (por ejemplo, un servicio proporcionado por el middleware) o una funcionalidad integrada en un componente.
- ✓ Es refinada en un atributo cuyo valor es sí o no, dependiendo de la presencia o no del mecanismo o dispositivo.

Confiabilidad.

Madurez. Esta sub-característica evidencia la capacidad del producto de software para evitar fallas, como resultado de las fallas en el software. Es refinada en un atributo Mean Time To Failure (MTTF) medido en el código fuente.

A nivel arquitectónico:

- ✓ El atributo se calcula mediante las siguientes mediciones:

$$\sum_i \text{madurez (componente } i) + \sum_j \text{madurez (conector } j).$$

Según los autores (Losavio, y otros, 2003) el atributo de madurez de los componentes COTS se conoce o se debería conocer.

Tolerancia a fallos. Esta sub-característica refleja la capacidad de mantener un determinado nivel de rendimiento en caso de fallos de software o de vulnerabilidad en su interfaz especificada.

A nivel arquitectónico:

- ✓ Significa contar con un mecanismo o dispositivo de software. Puede ser un componente independiente o una funcionalidad integrada en un componente, por ejemplo el manejo de excepciones o redundancia.
- ✓ Es refinada en un atributo cuyo valor es sí o no, dependiendo de la presencia o no del mecanismo o dispositivo.
- ✓ Puede ser refinado en un atributo cuyo valor se asocia con el mecanismo o dispositivo.

Recuperación. Esta sub-característica se expresa en tres elementos importantes, la capacidad para restablecer el nivel de rendimiento, la capacidad de recuperar los datos y el tiempo y el esfuerzo necesarios para ello.

A nivel arquitectónico:

- ✓ Esto significa la existencia de un mecanismo o dispositivo de software, que puede ser un componente o una parte de un componente, que permita restablecer el nivel de rendimiento y la recuperación de los datos, por ejemplo, la redundancia.
- ✓ Si el mecanismo existe, la recuperación se refina en el cálculo de los parámetros de los atributos relacionados con tiempo y esfuerzo. Debe ser calculado para cada componente, mecanismo o dispositivo.

Usabilidad.

Entendimiento. Se mide como la capacidad del producto de software para permitir al usuario entender si el software es adecuado, y cómo se puede utilizar para determinadas tareas y condiciones de uso.

Aprendizaje. Se expresa como la capacidad del producto de software para permitir al usuario aprender el funcionamiento del sistema.

Operabilidad. Esta sub-característica evidencia la capacidad del producto de software para permitir al usuario operar y controlar el sistema.

Estas sub-características pueden ser refinadas en los atributos asociados a los componentes de la interfaz gráfica del usuario (GUI , Graphic User Interface). Estas subcaracterísticas son independientes de la arquitectura, la cual es transparente a los usuarios, por lo que no se consideran en la adecuación del modelo.

Eficiencia.

Rendimiento. Refleja la capacidad del producto de software para proporcionar el tiempo de respuesta adecuado, tiempo de transformación y las tasas de rendimiento en el desempeño de su función bajo condiciones establecidas. Es un atributo que puede ser medido para cada funcionalidad del sistema.

A nivel arquitectónico:

- ✓ Se mide para cada función y cada funcionalidad que ejecute el usuario por medio de atributos calculados por las siguientes mediciones:

$$\Sigma_i \text{ Performance (funcionalidad de los componentes } i) + \Sigma_j \text{ Performance (conector } j)$$

Aprovechamiento de los recursos: Esta sub-característica manifiesta la cantidad y tipos de recursos utilizados en el desempeño de una función, así como la duración de esos usos. El atributo utilizado da una medida de la participación de los recursos utilizados. (espacio de los recursos utilizados y tiempo de utilización de los recursos).

A nivel arquitectónico:

- ✓ Los atributos pueden ser definidos y medidos para cada función, asociados al estilo arquitectónico utilizado. Espacio y tiempo están asociados a los componentes, y sus valores a cada componente y/o conector para cada una de las funciones.

Mantenibilidad.

Analizabilidad: Esta sub-característica mide la capacidad del producto de software de diagnosticar las deficiencias, posibles causas de fallos, o para identificar las partes que deben ser modificadas.

Facilidad para el cambio. Esta sub-característica se refleja en la capacidad del producto de software para permitir una modificación específica que debe aplicarse.

Estabilidad: Esta sub-característica percibe la capacidad del producto de software para evitar los efectos inesperados de las modificaciones del software, en especial el riesgo que provocan estas modificaciones.

Testeabilidad. Expresa la capacidad que presenta el producto de software de medir el esfuerzo necesario para validar el software modificado. Esta sub-característica es refinada en un atributo que mide la complejidad del código fuente, a través del cálculo de métricas de tamaño.

En el plano arquitectónico:

- ✓ Se tiene en cuenta la sub-característica de acoplamiento, que es una propiedad global de la arquitectura en relación con los intercambios entre los componentes, los atributos pueden ser medidos a través de métricas para cada componente en específico.
- ✓ Otra sub-característica es la modularidad, que expresa la topología de la arquitectura, como el número de componentes en función de un componente. Se trata de un atributo en el que se calcula la participación de cada componente, mediante métricas de tamaño.

Portabilidad

Adaptabilidad. Refleja la capacidad del producto de software para adaptarse a diferentes entornos especificados utilizando sólo su propia funcionalidad.

A nivel arquitectónico:

- ✓ La presencia de mecanismos de adaptación, por ejemplo, generalidad de parametrización
- ✓ Esta sub-característica es refinada en un atributo cuyo valor es sí o no, dependiendo de la presencia o no del mecanismo.

Capacidad de instalación. Esta sub-característica permite medir la capacidad del producto de software para ser instalado en un entorno determinado.

A nivel arquitectónico:

- ✓ La presencia de un mecanismo de instalación.
- ✓ Esta sub-característica es refinada en un atributo cuyo valor es sí o no, dependiendo de la presencia o no del mecanismo.

Coexistencia. Refleja la capacidad del producto de software para coexistir con otros programas informáticos independientes en un entorno común, compartiendo recursos comunes.

A nivel arquitectónico:

- ✓ La presencia de un mecanismo que facilite la coexistencia entre sistemas.
- ✓ Esta sub-característica es refinada en un atributo cuyo valor es sí o no, dependiendo de la presencia o no del mecanismo.

Reemplazo. Esta sub-característica mide la capacidad del producto de software de que se utilice en lugar de otro producto de software determinado para el mismo propósito en el mismo entorno. Se mide fundamentalmente la adaptabilidad y la capacidad de instalación.

A nivel arquitectónico:

- ✓ El atributo se expresa en una lista (nombre) de los componentes reemplazables, para cada componente.

A continuación se expone brevemente el resultado de la aplicación de este modelo de calidad adaptado para la arquitectura de software en un caso de estudio, donde se valoran dos arquitecturas candidatas, una que utiliza el patrón arquitectónico editor/suscriptor y otro que utiliza el patrón repositorio.

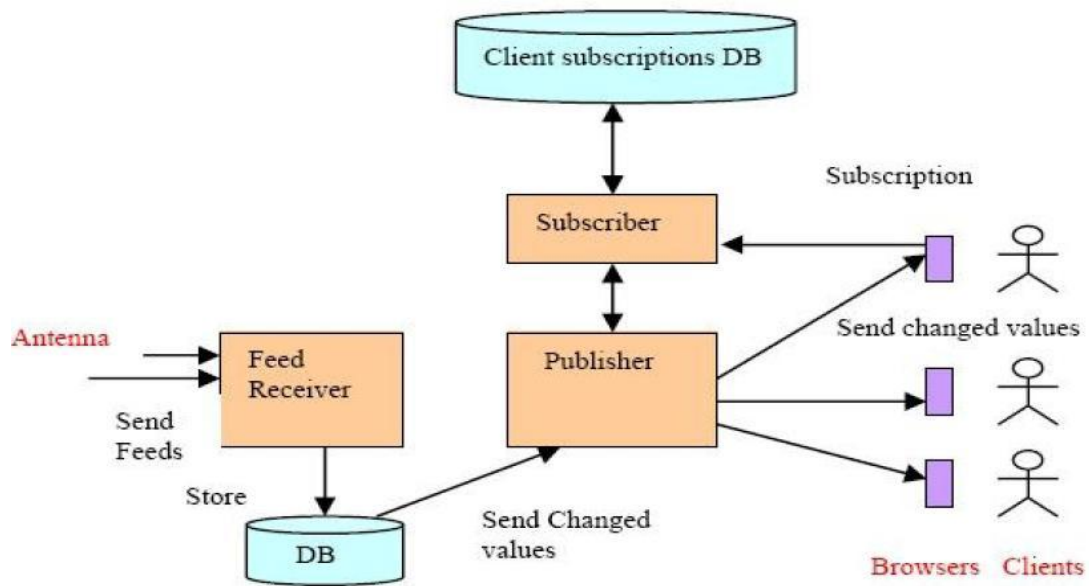


Fig. 4. Arquitectura basada en el patrón editor/suscriptor

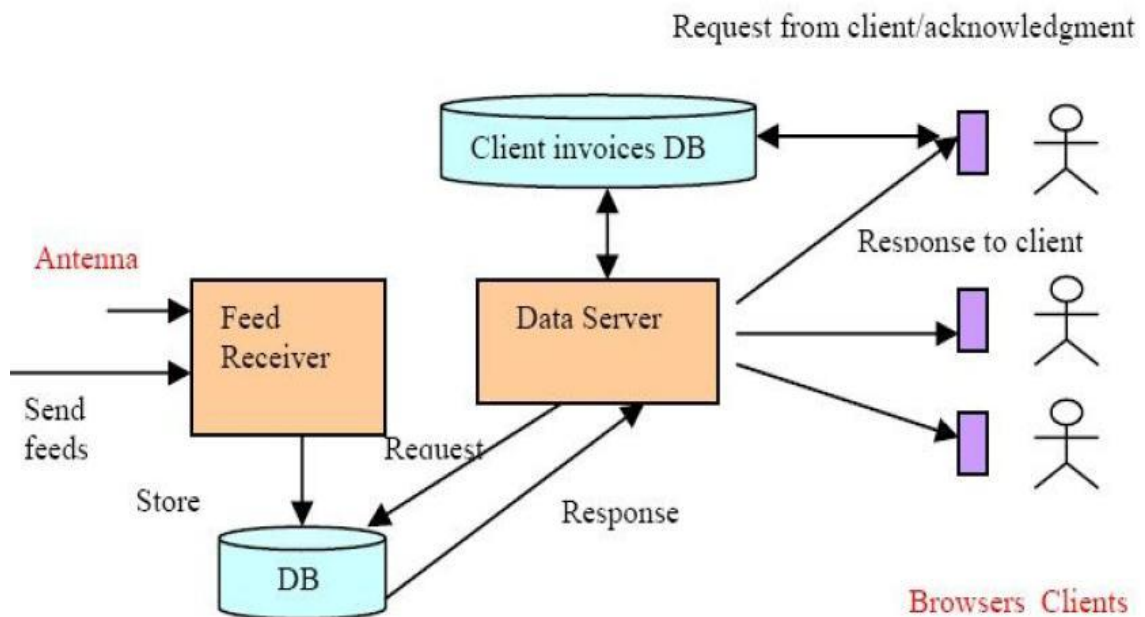


Fig. 5. Arquitectura basada en el patrón repositorio

En la siguiente tabla se presenta un resumen con los principales análisis de cada característica y sub-característica en las arquitecturas candidatas.

Característica	Sub-característica	Editor/Suscriptor	Repositorio	Comentarios y resultados
<i>Funcionalidad</i>	Idoneidad	si	si	
	Precisión	=	=	No es necesario un cálculo especial
	Interoperabilidad	si	si	Comunicación a través de los navegadores.
	Seguridad	Mecanismo de suscripción	Mecanismo para cada solicitud de cliente.	Editor/suscriptor es mejor.
<i>Confiabilidad</i>	Madurez	Madurez (Recepción) + Madurez (BD) + Madurez (Editor)+ Madurez (suscriptor) + Madurez (suscripción BD)	Madurez (Recepción) + Madurez (BD) + Madurez (el servidor de datos)+ Madurez (Peticiónes clientes a la BD)	Repositorio es mejor
	Tolerancia a fallos	=	=	Depende de mecanismos adicionales
	Recuperación	=	=	Depende de mecanismos adicionales
<i>Usabilidad</i>	-	=	=	Depende de los GUI de los navegadores.
<i>Eficiencia</i>	Comportamiento en el tiempo (tiempo transcurrido desde la recepción de datos hasta entrega de datos)	Tiempo (Recepción) + Tiempo (almacenamiento en la BD) + tiempo (enviar los cambios) + tiempo (Editor) + tiempo (enviar valores cambiados)	Tiempo (solicitud del cliente)+ Tiempo (peticiones del cliente a la BD) + Tiempo (Sevidor de datos) + Tiempo (Solicitud al Servidor) + Tiempo (BD)+ Tiempo (respuesta de la BD) + Tiempo (respuesta al cliente)	Editor/Suscriptor es mejor.
	Utilización de los recursos (tiempo)	El navegador muestra datos siempre.	El navegador solo muestra datos a solicitudes de	Repositorio es mejor

			los clientes	
	Utilización de los recursos (espacio)	Depende de las suscripciones de la BD.	Depende de las invocaciones a la BD.	Depende de cuestiones externas, como el volumen de solicitudes de cliente
<i>Mantenibilidad</i>	-	=	=	Depende del código de los componentes
<i>Portabilidad</i>	-	=	=	Depende de mecanismos adicionales

Tabla 5. Resumen de la utilización del modelo ISO 9126 adaptado a la arquitectura de software en un caso de estudio.

1.3.1.6 Consideraciones generales sobre los modelos de calidad.

Luego de un análisis detallado de cada uno de los modelos de calidad, se observa que de manera general todos abarcan los factores de calidad que influyen directamente en la calidad arquitectónica, funcionalidad, eficiencia, mantenibilidad y portabilidad.

El modelo McCall a pesar de realizar un refinamiento arbóreo de los factores de calidad que propone, las métricas son más bien de carácter general, muy difíciles de aplicar a la arquitectura de software como producto. Boehm por su parte realiza un análisis adicional acerca del rendimiento del hardware, tema muy relacionado con la eficiencia de la arquitectura diseñada, y mejora el refinamiento del factor mantenibilidad, pero al igual que su predecesor sus medidas no van a orientadas a la verificación de los atributos de calidad desde el diseño arquitectónico.

El metamodelo de Dromey, debido a su flexibilidad puede ser ajustado al caso específico de la arquitectura de software, transitando por los cinco pasos que propone. Haciendo énfasis en los pasos dos y tres donde se especifican los distintos componentes que conforman el producto, y la determinación y categorización de sus principales implicaciones en la calidad del producto. Sin embargo, no existen precedentes investigaciones que usen esta metodología propuesta por Dromey en el desarrollo de un modelo de calidad enfocado a la arquitectura de software.

El modelo de FURPS realiza como aporte la adición de las restricciones de diseño, las cuales establecen el diseño del sistema; los requerimientos de interfaz, que especifican el comportamiento de los elementos externos con los que el sistema debe interactuar; y por último, los requerimientos físicos, que definen ciertas

propiedades que el sistema debe poseer, en términos de materiales, forma, peso, tamaño (por ejemplo, requisitos de hardware, configuración de red). Todos estos factores que tiene presente este modelo son relevantes en el análisis de la calidad arquitectónica, pues interactúan directamente con el establecimiento de la arquitectura de software.

Sin embargo, es sin duda la adecuación del estándar ISO 9126, realizado por la Dr. Francisca Losavio, para el caso específico de la arquitectura de software como producto intermedio del ciclo de desarrollo (Losavio, y otros, 2003), el modelo que más elementos aporta al análisis de la calidad arquitectónica. No solo se analizan cada una de las 6 características que propone la norma ISO/IEC 9126-1 (ISO/IEC-9126-1, 2001) y sus sub-características desde un punto de vista arquitectónico, sino que facilita la comprensión acerca de cuáles elementos debe tener una arquitectura de software para lograr valores adecuados de cada atributo de calidad.

A pesar de que los modelos de calidad logren medir el grado de cumplimiento de los factores de calidad en un producto específico, la labor de la evaluación de la arquitectura de software no abarca solamente ese marco de medición.

Evaluar una arquitectura es verificar el cumplimiento de los atributos de calidad, pero es además llevar a cabo un proceso de priorización de características de calidad en caso del que modelo no pueda ser conclusivo ante arquitecturas candidatas, es también valorar mediante escenarios que elemento arquitectónico debe ser cambiado para mejorar atributos de rendimiento o mantenibilidad. Y luego de todo esto, es el aporte que debe realizar un proceso de evaluación arquitectónica al equipo de desarrollo, logrando un intercambio entre el equipo de desarrollo e involucrados que enriquezca el diseño arquitectónico y beneficie de manera directa las etapas posteriores del ciclo de desarrollo. Es por eso que a continuación analizaremos un conjunto de técnicas y métodos que separan un poco el concepto de calidad de la evaluación arquitectónica y va más enfocado a la arquitectura de software como disciplina.

1.4 Evaluación de arquitecturas de software.

La arquitectura de software es uno de los elementos del diseño de un sistema que más influye en el éxito del proyecto de desarrollo, por lo que es muy importante

estar en capacidad de tomar decisiones acertadas sobre ella, en diversos tipos de situaciones, entre las cuales destacan (Bass, y otros, 2005)

- ✓ Comparación de alternativas similares.
- ✓ Comparación de la arquitectura original y la modificada.
- ✓ Comparación de la arquitectura de software con respecto a los requerimientos del sistema
- ✓ Comparación de una arquitectura de software con una propuesta teórica.
- ✓ Valoración de una arquitectura en base a escalas específicas.

Evaluar una arquitectura, consiste en evaluar el potencial de la arquitectura diseñada, para lograr altos niveles de rendimiento del software, o para cumplir restricciones de hardware, software o de interfaz gráfica, exigidas por los clientes o un mercado específico. Además, mediante la arquitectura de software es posible determinar la estructura del proyecto de desarrollo del sistema, sobre elementos como el control de configuración, calendarios, control de recursos, metas de desempeño, estructura del equipo de desarrollo y otras actividades que se realizan con la arquitectura del sistema como apoyo principal. En este sentido, la garantía de una arquitectura correcta, cumple un papel fundamental en el éxito general del proceso de desarrollo y la calidad del mismo. (Kazman, 2006)

La evaluación de la arquitectura de un sistema en las fases tempranas del desarrollo traen un conjunto de beneficios, entre lo que se pueden mencionar:

- ✓ *Beneficios financieros:* Al evaluarse la arquitectura se predicen un conjunto de decisiones de diseño, que provocan disminución del tiempo, y de recursos, lo que esto actúa de forma directa en la disminución de los costos de desarrollo.
- ✓ *Mejora en la documentación de la arquitectura.* Al evaluarse la arquitectura, es necesario la claridad en un conjunto de elementos de estructura organizacional, definición de escenarios, atributos de calidad relevantes, restricciones en la implementación, entre otros, que al ser analizados, obliga de cierta manera, un alto nivel de documentación sobre estos componentes.

- ✓ *Valida requerimientos:* Al evaluarse si la arquitectura cumple con los requerimientos, siempre salen discusiones sobre estos, que permiten una mayor claridad por parte de los involucrados, y mejor precisión y orden de prioridad de cada requerimiento.
- ✓ *Mejora la arquitectura:* Basado en los errores identificados, se rediseña el conjunto de elementos arquitectónicos, se valoran las decisiones tempranas en el diseño y se establece una visión de la arquitectura durante las fases posteriores.
- ✓ *Aprendizaje organizacional:* De todo el proceso, donde participan de forma activa los involucrados en el desarrollo, se logra un trabajo en equipo que fortalece la capacidad de sinergia entre los distintos actores del sistema, y la posibilidad de localizar y solucionar problemas en el futuro.

Por todo lo antes expuestos se hace necesarios realizar un estudio de las técnicas y métodos existentes hasta el momento que permiten una evaluación técnica de la arquitectura diseñada para un sistema.

1.4.1 Técnicas de evaluación de arquitecturas de software.

Según (Bosch, 2000), las técnicas utilizadas para la evaluación de la arquitectura requieren grandes esfuerzos por parte del ingeniero de software para crear especificaciones y predicciones. Estas técnicas requieren información del sistema a desarrollar que no está disponible durante el diseño arquitectónico, sino al principio del diseño detallado del sistema.

En vista de que el interés es tomar decisiones de tipo arquitectónico en las fases tempranas del desarrollo, son necesarias técnicas que requieran poca información detallada y puedan conducir a resultados relativamente precisos (Bosch, 2000). Las técnicas existentes en la actualidad para evaluar arquitecturas permiten hacer una evaluación cuantitativa sobre los distintos niveles arquitectónicos, pero se tienen pocos medios para predecir el máximo (o mínimo) teórico para las arquitecturas de software. Sin embargo, debido al costo de realizar este tipo de evaluación, en muchos casos los arquitectos de software evalúan cualitativamente, para decidir entre las alternativas de diseño (Bosch, 2000). Bosch propone diferentes técnicas de evaluación de arquitecturas de software, a saber: evaluación basada en escenarios,

evaluación basada en simulación, evaluación basada en modelos matemáticos y evaluación basada en experiencia.

1.4.1.1 Evaluación basada en escenarios.

De acuerdo con (Kazman, y otros, 2003), un escenario es una breve descripción de la interacción entre los involucrados y el sistema. Por ejemplo, un usuario hará la descripción en términos de la ejecución de una tarea; un encargado de mantenimiento hará referencia a cambios que deban realizarse sobre el sistema; un desarrollador se enfocará en el uso de la arquitectura para efectos de su construcción o predicción de su desempeño.

Según (Kazman, y otros, 2003) 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 factor de calidad relacionado con el escenario que se analiza.

Los escenarios proveen un vehículo que permite concretar y entender atributos de calidad. Un conjunto de autores (Kazman, y otros, 2003) y (Carriere, y otros, 2007) coinciden en la importancia del uso de los escenarios, no sólo para efectos de la evaluación de arquitecturas de software.

Entre las ventajas de su uso están:

- ✓ Son simples de crear y entender.
- ✓ Son poco costosos y no requieren mucho entrenamiento.
- ✓ Son efectivos.

Actualmente las técnicas basadas en escenarios cuentan con dos instrumentos de evaluación relevantes, a saber: el Utility Tree propuesto por (Kazman, y otros, 2003) y los Profiles, propuestos por (Bosch, 2000).

1.4.1.2 *Evaluación basada en simulación.*

Bosch (Bosch, 2000) establece que la evaluación basada en simulación utiliza una implementación de alto nivel de la arquitectura de software. El enfoque básico consiste en la implementación de componentes de la arquitectura y la implementación a cierto nivel de abstracción- del contexto del sistema donde se supone va a ejecutarse. La finalidad es evaluar el comportamiento de la arquitectura bajo diversas circunstancias. Una vez disponibles estas implementaciones, pueden usarse los perfiles respectivos para evaluar los atributos de calidad.

El proceso de evaluación basada en simulación sigue los siguientes pasos (Bosch, 2000):

- ✓ *Definición e implementación del contexto.* Consiste en identificar las interfaces de la arquitectura de software con su contexto, y decidir cómo será simulado el comportamiento del contexto en tales interfaces.
- ✓ *Implementación de los componentes arquitectónicos.* La descripción del diseño arquitectónico debe definir, por lo menos, las interfaces y las conexiones de los componentes, por lo que estas partes pueden ser tomadas directamente de la descripción de diseño. El comportamiento de los componentes en respuesta a eventos sobre sus interfaces puede no ser especificado claramente, aunque generalmente existe un conocimiento común y es necesario que el arquitecto lo interprete, por lo que éste decide el nivel de detalle de la implementación.
- ✓ *Implementación del perfil.* Dependiendo del atributo de calidad que el arquitecto de software intenta evaluar usando simulación, el perfil asociado necesitará ser implementado en el sistema. El arquitecto de software debe ser capaz de activar escenarios individuales, así como también ejecutar un perfil completo usando selección aleatoria, basado en los pesos normalizados de los mismos.
- ✓ *Simulación del sistema e inicio del perfil.* El arquitecto de software ejecutará la simulación y activará escenarios de forma manual o automática, y obtendrá resultados de acuerdo al atributo de calidad que está siendo evaluado.

- ✓ *Predicción de atributos de calidad.* Dependiendo del tipo de simulación y del atributo de calidad evaluado, se puede disponer de cantidades excesivas de datos, que requieren ser condensados. Esto permite hacer conclusiones acerca del comportamiento del sistema.

En términos de los instrumentos asociados a las técnicas de evaluación basadas en simulación, se encuentran los lenguajes de descripción arquitectónica y los modelos de colas.

Esta técnica presenta un grupo de dificultades, como son, la exactitud de los resultados de la evaluación depende, a su vez, de la exactitud del perfil utilizado para evaluar el atributo de calidad y de la precisión con la que el contexto del sistema simula las condiciones del mundo real. Estos elementos tributan a que la evaluación no tenga un nivel alto de precisión, pues sus resultados depende de factores subjetivos, que pueden ser logrados o no. Además esta técnica es utilizada para evaluar solo requerimientos de calidad operacional, como desempeño y confiabilidad.

1.4.1.3 Evaluación basada en modelos matemáticos.

Bosch (Bosch, 2000) establece que la evaluación basada en modelos matemáticos se utiliza para evaluar atributos de calidad operacionales. Permite una evaluación estática de los modelos de diseño arquitectónico, y se presentan como alternativa a la simulación, dado que evalúan el mismo tipo de atributos. Ambos enfoques pueden ser combinados, utilizando los resultados de uno como entrada para el otro.

El proceso de evaluación basada en modelos matemáticos sigue los siguientes pasos (Bosch, 2000):

- ✓ *Selección y adaptación del modelo matemático.* La mayoría de los centros de investigación orientados a atributos de calidad han desarrollado modelos matemáticos para medir sus atributos de calidad, los cuales tienden a ser muy elaborados y detallados, así como también requieren de cierto tipo de datos y análisis. Parte de estos datos requeridos no están disponibles a nivel de arquitectura, y la técnica requiere mucho esfuerzo para la evaluación arquitectónica, por lo que el arquitecto de software se ve obligado a adaptar el modelo.

- ✓ *Representación de la arquitectura en términos del modelo.* El modelo matemático seleccionado y adaptado no asume necesariamente que el sistema que intenta modelar consiste de componentes y conexiones. Por lo tanto, la arquitectura necesita ser representada en términos del modelo.
- ✓ *Estimación de los datos de entrada requeridos.* El modelo matemático aún cuando ha sido adaptado, requiere datos de entrada que no están incluidos en la definición básica de la arquitectura. Es necesario estimar y deducir estos datos de la especificación de requerimientos y de la arquitectura diseñada.
- ✓ *Predicción de atributos de calidad.* Una vez que la arquitectura es expresada en términos del modelo y se encuentran disponibles todos los datos de entrada requeridos, el arquitecto está en capacidad de calcular la predicción resultante del atributo de calidad evaluado.

Entre los instrumentos que se cuentan para las técnicas de evaluación de arquitecturas de software basada en modelos matemáticos, se encuentran las Cadenas de Markov y los Reliability Block Diagrams.

Entre las desventajas que presenta esta técnica se encuentra la inexistencia de modelos matemáticos apropiados para los atributos de calidad relevantes (Bosch, 2000), y el hecho de que el desarrollo de un modelo de simulación completo puede requerir esfuerzos sustanciales.

1.4.1.4 Evaluación basada en experiencia.

Bosch (Bosch, 2000) establece que en muchas ocasiones los arquitectos e ingenieros de software otorgan valiosas ideas que resultan de utilidad para la evasión de decisiones erradas de diseño. Aunque todas estas experiencias se basan en evidencia anecdótica; es decir, basada en factores subjetivos como la intuición y la experiencia. Sin embargo, la mayoría de ellas puede ser justificada por una línea lógica de razonamiento, y pueden ser la base de otros enfoques de evaluación (Bosch, 2000)

Existen dos tipos de evaluación basada en experiencia: la evaluación informal, que es realizada por los arquitectos de software durante el proceso de diseño, y la realizada por equipos externos de evaluación de arquitecturas.

Estos tipos de técnicas de evaluación, no están basadas en conceptos teóricos, sino que se apoyan en factores intrínsecos de las personas, que pueden ser efectivos o no. De manera que, basar el aseguramiento del logro de los atributos de calidad en este tipo de técnicas, es no confiable para el correcto desarrollo del producto de software.

La tabla 7 presenta de forma resumida los instrumentos utilizados por las diferentes técnicas de evaluación.

Técnicas de evaluación	Instrumentos de evaluación
Basada en escenarios	<ul style="list-style-type: none"> ✓ Perfiles. ✓ Árbol de utilidad.
Basada en simulación	<ul style="list-style-type: none"> ✓ Lenguajes de descripción arquitectónica (ADLs) ✓ Modelos de colas.
Basadas en modelos matemáticos	<ul style="list-style-type: none"> ✓ Cadenas de Markov ✓ Reliability Block Diagrams
Basadas en experiencias	<ul style="list-style-type: none"> ✓ Intuición y experiencia ✓ Tradición ✓ Proyectos similares

Tabla 6. Instrumentos asociados a las distintas técnicas de evaluación de arquitecturas de software.

1.4.1.5 Consideraciones generales sobre las técnicas de evaluación de arquitectura.

Las técnicas de evaluación de arquitectura de software propuestas por Bosch, de manera general, logran cierta evaluación en las fases tempranas del ciclo de desarrollo. A través de análisis de simulación y de escenarios, permiten al equipo de evaluación, no solo evaluar un conjunto de atributos de calidad de manera cualitativa, sino tomar decisiones tempranas en el diseño.

Sin embargo, su mayoría no hacen énfasis en todos los atributos de calidad relevantes para la arquitectura, y basan sus resultados en actividades que no pueden ser altamente medibles, pues depende en cierta medida de un factor subjetivo, como lo es, la representatividad de los escenarios, la exactitud del perfil seleccionado para cada atributo, la precisión con la que el contexto del sistema simula las condiciones del mundo real, la intuición y la experiencia.

La principal dificultad que presenta este conjunto de técnicas es que la selección de los atributos de calidad a medir se establece a consideración de la característica

de la técnica a usar, como es el caso de los atributos de calidad operacionales en las técnicas basadas en modelos matemáticos; o de manera empírica, a consideración del equipo de desarrollo y los involucrados, lo que provoca que los equipos con poca experiencia no logren un análisis profundo de la evaluación arquitectónica. En este caso el establecimiento de un modelo que guíe el refinamiento de las medidas de calidad es de suma importancia para el éxito del proceso evaluativo.

1.4.2 Métodos de evaluación de arquitecturas de software.

Los métodos han sido otras de las herramientas desarrolladas con el objetivo de evaluar una arquitectura de software.

De acuerdo con (Kazman, y otros, 2005) hasta hace poco no existían métodos de utilidad general para evaluar arquitecturas de software. Si alguno existía, sus enfoques eran incompletos, ad hoc, y no repetibles, lo que no brindaba mucha confianza. En virtud de esto, múltiples métodos de evaluación han sido propuestos. A continuación se explican algunos de los más importantes.

1.4.2.1 Método de análisis de arquitectura de software (Software Architecture Analysis Method, SAAM).

Contexto.

El método SAAM constituye el primer método de evaluación de arquitectura ampliamente promulgado. Este fue creado para evaluar la modificabilidad de la arquitectura en todo su alcance.

Objetivo general.

El método SAAM tiene como objetivo evaluar un grupo de atributos de calidad que debe lograr la arquitectura de software, a través del uso de escenarios. En la práctica ha demostrado ser útil para la rápida evaluación de atributos de calidad como: modificabilidad, portabilidad y extensibilidad. En resumen se puede afirmar, que al evaluar una arquitectura, SAAM indica los puntos de fortalezas y debilidades, junto con los puntos de la arquitectura que no cumple con los requisitos de modificabilidad.

Factores claves para su desarrollo.

El desarrollo de este método fue motivado por una variedad de opiniones acerca de la arquitectura de software, la ausencia de métodos y las pautas comunes para hacer frente a esta actividad. Otra de las causas fue la imposibilidad que tenían los diseñadores y los arquitectos de los sistemas de predecir el comportamiento de la calidad de software, ni tan siquiera en la etapa de implementación.

Requisitos previos y entradas.

Los requisitos fundamentales para llevar a cabo la evaluación a través de SAAM, es el establecimiento de los atributos de calidad a ser evaluados. Esto impone la adopción de escenarios como medios para describir, especificar y evaluar los atributos. Además de estos elementos, debe estar disponible para todos los participantes la descripción de la arquitectura del sistema. Una serie de escenarios que describan la interacción de un usuario con el sistema, son las principales entradas para una sesión de evaluación SAAM.

Pasos a realizar para la evaluación.

El método SAAM consta de seis pasos principales, que generalmente van precedidas de una breve visión general del negocio y el contexto en que el sistema debe cumplir cada una de sus funcionalidades.

Paso 1. Desarrollar escenarios: El primer paso en un período de sesiones SAAM es un ejercicio de lluvia de ideas con el objetivo de determinar las actividades que describen de forma completa el comportamiento de un atributo de calidad. Estas actividades, junto con las posibles modificaciones que los interesados pueden anticipar, se agrupan en el llamado sistema de escenarios. En la elaboración de escenarios, el desafío consiste en capturar las principales funcionalidades, usuarios y el nivel mínimo que debe alcanzar cada atributo ante previsible cambios del sistema. Este ejercicio se realiza generalmente dos veces. Cuantas más interacciones, mayor es la información arquitectónica que se comparte, y por tanto más escenarios se identifican por lo participantes. La recomendación es llevar a cabo estas actividades en paralelo.

Paso 2. Describir la arquitectura: En la segunda etapa del período evaluativo se presentan la arquitectura candidata. Las notaciones de la arquitectura presentada deben ser bien comprendidas por lo participantes y deberá indicar la representación estática del sistema (componentes, sus interconexiones y la relación con el medio donde se desarrollan) y su comportamiento dinámico.

Paso 3. Clasificar y Priorizar Escenarios: En este punto se clasifican los escenarios en directos e indirectos. Los escenarios directos son los que están relacionados con el cumplimiento de uno o más atributos de calidad. Los escenarios indirectos es la secuencia de eventos para lograr que la arquitectura sufra los menores cambios posibles durante el desarrollo del sistema. La priorización de los escenarios se basa en un proceso de votación. Como SAAM aborda fundamentalmente la evaluación de la modificabilidad del sistema, el resultado de la votación lo constituyen un conjunto de escenarios indirectos con mayor probabilidad de efectuarse.

Paso 4. Evaluar los escenarios indirectos: En el caso de un escenario directo el arquitecto demuestra cómo el escenario sería ejecutado por la arquitectura. En el caso de un escenario indirecto el arquitecto describe la forma en que la arquitectura debe ser modificada para lograr el escenario. En cada uno de estos escenarios indirectos se debe haber identificado las modificaciones arquitectónicas necesarias para facilitar el escenario, junto con los involucrados, los nuevos componentes del sistema, el costo estimado y el esfuerzo para aplicar la modificación.

Paso 5. Evaluar la interacción de escenarios: Cuando dos o más escenarios afectan al mismo componente de la arquitectura, se dice que interactúan. En este caso, los componentes implicados deben ser modificados o divididos en sub-componentes, a fin de evitar la interacción de los diferentes escenarios.

Paso 6. Crear una evaluación global: Durante este paso se le asigna un peso a cada uno de los escenarios en correspondencia con su importancia para el éxito del sistema. Se realiza una ponderación de estos pesos, unido a criterio como los costos, riesgos, tiempo de salida al mercado del producto, entre otros. Sobre la base de esta ponderación se propone la arquitectura más adecuada, cuando se valoran distintas alternativas. Alternativas para la arquitectura correcta puede ser además aquella que abarque la mayoría de los escenarios directos y los escenarios indirectos que requieran de menos esfuerzo ante una modificación de la arquitectura.

Funciones de los involucrados.

Durante el proceso de evaluación de SAAM quedan identificados tres roles fundamentales:

Involucrados externos: Este grupo está formado por las personas interesadas que no tienen participación directa en el proceso de desarrollo de arquitectura de software. Su función es presentar en las sesiones de evaluación los objetivos del negocio, el sistema de atributos de calidad y su nivel de progreso a lo largo del ciclo de desarrollo, así como proporcionar los escenarios directos e indirectos junto con su prioridad y clasificación. Ejemplos de involucrados externos son: los clientes, usuarios finales, especialistas en marketing, los administradores de sistemas, etc.

Involucrados internos: Son las personas que tienen una participación directa en la propuesta de estrategias de arquitectura de software que puedan satisfacer los requisitos de calidad. Tienen la función de analizar, definir y presentar los conceptos arquitectónicos y la estimación de los costes asociados a la programación de estas estrategias. Ejemplos de los involucrados internos son los arquitectos de software, analistas de sistemas o el equipo de arquitectura.

Equipo de evaluación SAAM: Las personas que conforman este equipo no tienen ningún interés directo en la definición de arquitectura de software, pero lleva a cabo la sesión de evaluación SAAM. Tienen la función de apoyar a los actores del sistema en la presentación de los objetivos del negocio, así como en la presentación de los atributos de calidad a evaluar, y la definición de los escenarios. El equipo de evaluación consta de un jefe de equipo o portavoz, el grupo de expertos internos y externos, que establecen los pesos de cada escenario definido, y un secretario.

Esfuerzo necesario para su aplicación.

El esfuerzo realizado por el equipo de evaluación SAAM depende en gran medida del tamaño del proyecto y el número de atributos de calidad que debe cumplimentar. Una sesión abarca los 6 pasos planteados con anterioridad, que deben realizarse en un día. Esto excluye el tiempo de preparación y esfuerzo del arquitecto en la preparación de la descripción de la arquitectura y los escenarios de evaluación. Dependiendo del número de involucrados internos y externos, la duración del período de sesiones varía también.

Herramientas de apoyo.

Hasta el momento no son necesarias herramientas de apoyo para llevar a cabo las sesiones de evaluación de SAAM. La realización del proceso de votación para establecer el peso de cada escenario y la estimación de costo y esfuerzo en las posibles modificaciones de la arquitectura son las únicas técnicas utilizadas.

Alternativas a su utilización.

Tomando como punto de referencia el SAAM, se han creado varios métodos capaces de evaluar el atributo de modificabilidad. Uno de ellos es el ATAM (Architecture Trade-off Analysis Method), desarrollado por el mismo grupo que inició SAAM. (Referencia 2) Otro a nombrar es el método ALMA (Architecture Level Modifiability Analysis) (referencia 5 y 6), el cual abarca también el atributo de calidad de modificabilidad, como centro de su evaluación.

Fortalezas del SAAM.

- ✓ Las principales fortalezas de la utilización del método SAAM para la evaluación arquitectónica son:
- ✓ Aumento de la comprensión por los involucrados de la arquitectura que está siendo analizada.
- ✓ Luego de una sesión SAAM, la documentación de la arquitectura es refinada.
- ✓ Mejora la comunicación entre las partes interesadas.

Con respecto al análisis del atributo de modificabilidad sus fortalezas viene dadas por:

- ✓ Se realiza un análisis de los escenarios desde la perspectiva arquitectónica de futuros cambios en el sistema.
- ✓ Se identifican zonas con un alto nivel de complejidad.
- ✓ Se estiman los costos y esfuerzos para realizar cambios necesarios.
- ✓ Basándose en los análisis de efectos secundarios ante los distintos cambios, es posible la creación de planes de trabajo para el desarrollo futuro del sistema.

Comentarios generales.

Luego del estudio del método SAAM se han identificado un conjunto de reflexiones que pueden abordar aspectos para futuras mejoras.

- ✓ El proceso de generación de escenarios se basa en una visión de los involucrados externos. Es muy difícil ver por ejemplo a un actor en la definición de un escenario indirecto.

- ✓ El método no ofrece un claro indicador de la calidad arquitectónica en los atributos que analiza.
- ✓ En la descripción de la arquitectura no se establece una notación ni aspectos esenciales que logre que esta actividad sea entendible por los involucrados externos.
- ✓ El equipo de evaluación se basa únicamente en las distintas alternativas que proponen los arquitectos. No hay capacidad para valorar otras opciones arquitectónicas por parte del equipo de evaluación, pues no está familiarizado con las metas del negocio.

1.4.2.2 Método de análisis de interacciones de la arquitectura de software. (Architecture Trade-off Analysis Method, ATAM)

Contexto.

El desarrollo del método ATAM se realizó sobre la base del SAAM. El propósito general de su diseño fue permitir la evaluación de un mayor número de atributos. Otro de los aspectos que abarca ATAM, que no se encuentra reflejado en su antecesor, es la exploración de las interdependencias que se establecen entre los distintos atributos.

Objetivo general.

El método ATAM analiza cómo la arquitectura del software satisface los atributos de calidad como: modificabilidad, portabilidad, extensibilidad e integrabilidad. Además proporciona información sobre el cumplimiento de cada atributo de calidad teniendo en cuenta las interdependencias que se establecen entre ellos.

Factores claves para su desarrollo.

El diseño del método ATAM ha sido motivado por aspectos como:

- ✓ Los involucrados externos al desarrollo del software consideran todos los cambios arquitectónicos como igualmente posibles.
- ✓ Las metas del negocio y los aspectos técnicos de la arquitectura eran discutidos en la misma sesión de evaluación.
- ✓ La falta de conciencia en las partes interesadas de que los riesgos arquitectónicos, a largo plazo, ponen en peligro el cumplimiento de las metas del negocio.

- ✓ Ausencia de métodos de evaluación que consideraran el impacto de las decisiones de arquitectura en el logro de atributos de calidad como: disponibilidad, rendimiento, seguridad, modificabilidad, facilidad de uso, el tiempo de comercialización, etc.

Requisitos previos y entradas.

Para realizar con éxito una sesión de evaluación ATAM, los participantes deben tener en cuenta un conjunto de requisitos previos:

- ✓ *Los evaluadores deben comprender la arquitectura del sistema.* Reconocer los parámetros de la arquitectura, definir sus implicaciones con respecto a los atributos de calidad del sistema, y comparar estas implicaciones con el cumplimiento de las funcionalidades del sistema.
- ✓ *Identificar cuidadosamente los puntos de sensibilidad, los puntos de desventajas y los puntos de riesgos.* Un punto de sensibilidad está dado por uno o varios componentes que son críticos en el diseño arquitectónico para el logro de cierto atributo de calidad. Un punto de desventaja es un punto de sensibilidad crítico para el logro de múltiples atributos de calidad. Los puntos de riesgos son un subconjunto de los puntos de sensibilidad que provocan la imposibilidad del logro de las metas de calidad del software.
- ✓ *Lograr un entendimiento de cómo la arquitectura logra el cumplimiento de los atributos de calidad.* Esto puede lograrse a través de la descripción detallada de escenarios tipos.

Pasos a realizar para la evaluación.

El método ATAM consta para su implementación de cuatro fases: presentación, investigación y análisis, pruebas y presentación de informes. Cada fase es una colección de pasos. La fase de presentación implica el intercambio de información a través de presentaciones. La de investigación y la fase de análisis se refiere a la evaluación de los principales atributos de calidad teniendo en cuenta los requisitos que debe cumplir el sistema y los enfoques arquitectónicos. En la fase de prueba se comparan los resultados de la fase anterior con las necesidades de las partes interesadas. Por último, en la fase de presentación de informe, se resumen los

resultados de la evaluación ATAM. A continuación se presentan los pasos en detalle para la evaluación a través de ATAM.

Fase de presentación de ATAM.

Paso 1. Presentación del método: El líder del grupo de evaluación realiza una breve descripción de cómo se llevará a cabo el proceso evaluativo. Esta actividad se hace con el objetivo de fijar expectativas en los participantes y aclarar las dudas que puedan tener la respecto.

Paso 2. Presentar el negocio: Un portavoz del proyecto describe los objetivos que motivan el desarrollo del producto, y por tanto los aspectos que constituyen la guía para el logro de la calidad arquitectónica.

Paso 3. Presentar la arquitectura: Durante este paso el arquitecto describe la arquitectura de software del sistema, centrándose en la forma en que aborda el negocio, teniendo en cuenta los principales elementos de la presentación anterior.

Fase de Investigación y Análisis de ATAM.

Paso 4. Identificar los enfoques arquitectónicos: En este paso se identifica los enfoques y estilos arquitectónicos, pero no son analizados aún.

Paso 5. Generar el árbol de utilidad del atributo de calidad: Se identifican los atributos que conforman el árbol de utilidad y se especifican los escenarios priorizados para el cumplimiento de los distintos niveles del árbol, teniendo en cuenta los estímulos y la respuesta de cada uno.

Paso 6. Analizar los enfoques arquitectónicos: Sobre la base de los escenarios definidos anteriormente y la arquitectura presentada, se identifican los riesgos potenciales, los puntos de sensibilidad y los puntos de desventaja de cada uno de los componentes arquitectónicos. Se establecen además las fortalezas de la arquitectura ante el cumplimiento de atributos de calidad específicos.

Fase de Pruebas de ATAM.

Paso 7. Lluvia de ideas y dar prioridad a los escenarios: Durante una sesión de lluvias de ideas se establecen un amplio grupo de escenarios. En esta sesión participan las partes interesadas y junto al equipo priorizan los escenarios a mediante votación.

Paso 8. Reanálisis de los enfoques arquitectónicos: En este paso se toma como punto de partida el análisis del paso 6, lo que ya valorando con mayor precisión los

escenarios priorizados en el paso anterior. El objetivo es identificar y documentar cualquier otro enfoque arquitectónico que pueda ser generado luego de establecer la prioridad en los escenarios.

Fase de presentación de informes de ATAM.

Paso 9. Presentación de los resultados: En este último paso, sobre la base de la información recogida durante las tres fases anteriores, el equipo de evaluación presenta un informe con las conclusiones a las partes interesadas. Entre los elementos que debe abarcar el informe están: las medidas y decisiones arquitectónicas en cada uno de los pasos, los estilos arquitectónicos documentados, la última versión de los escenarios con sus prioridades, los atributos que conforman el árbol de utilidad, así como los puntos de riesgos, de sensibilidad y los puntos de desventajas.

Funciones de los involucrados.

Al igual que el SAAM, el ATAM establece tres clasificaciones de roles que participan en el proceso de evaluación, los interesados internos, los interesados externos y el equipo de evaluación.

Involucrados externos: Este grupo no está implicado directamente con el proceso de desarrollo de la arquitectura de software. Durante el período de sesiones su función es presentar el contexto empresarial del proyecto, sobre la base de las necesidades iniciales para ofrecer escenarios. Ejemplos de involucrados externos son: los clientes, los usuarios finales, administradores de sistemas, patrocinadores, etc.

Involucrados internos: Este personal está directamente vinculado en el proceso de desarrollo de la arquitectura de software. Su función es analizar, definir e implementar la arquitectura. Durante la sesión de evaluación ATAM, son los responsables de describir, presentar y evaluar, junto con el equipo de ATAM, la arquitectura de software. Ejemplos de involucrados internos son: los arquitectos, los líderes del equipo de diseño, el responsable de realizar las pruebas al producto, y los integradores.

Equipo de evaluación ATAM: Este equipo está conformado por personas externas la grupo de desarrollo, que mantenga una posición neutral ante las decisiones. Estas personas no deben tener un interés directo con el desarrollo del

producto, solamente son invitados a dirigir las sesiones de evaluación. Es responsabilidad además de este grupo, rectorar el proceder de la evaluación, registro de toda la documentación necesaria y la presentación de los resultados finales. Este grupo por lo general cuenta con un jefe de equipo o portavoz, analista de la arquitectura y el secretario.

Esfuerzo necesario para su aplicación.

Dado que el método es muy dependiente del desarrollo de los proyectos, no existe ninguna cifra real del esfuerzo a realizar en una evaluación de ATAM. La única referencia está dada por las fases del método, que se distribuyen en tres días diferentes. Después de cada fase, el equipo de evaluación ATAM, realiza un receso para estructurar y organizar la información, y preparar la próxima fase. Teniendo en cuenta el número de participantes, en el período de sesiones, el número de atributos de calidad que deben evaluarse, así como el tamaño del proyecto y la complejidad de la arquitectura se pueden estimar cifras de esfuerzo a realizar.

Herramientas de apoyo.

Recientemente, se ha desarrollado una herramienta, que apoya integralmente una sesión de evaluación de ATAM. La herramienta ha sido desarrollada por Stephan Kurpjuviet, a través de una tesis doctoral en el Instituto Fraunhofer, en colaboración con el Software Engineering Institute (SEI) (Referencia número 10). La herramienta proporciona un conjunto de elementos como: la documentación de la evaluación, la gestión y control de las distintas versiones de la arquitectura, las limitaciones de los modelos de datos, las transformaciones arquitectónicas de apoyo, las restricciones definidas por los usuarios, entre otras funcionalidades. Esta herramienta no describe el proceso de evaluación de ATAM y aún no ha sido probada en un proyecto real, por lo que no se puede valorar la eficacia de su utilización.

Alternativas.

La alternativa más cercana al método ATAM, es el ALMA. El ALMA logra evaluar el riesgo de una modificación arquitectónica al mismo nivel que lo logra ATAM, sin embargo el análisis de las interrelaciones entre los atributos de calidad, no existe aún un método que logre un nivel de especificación en este sentido como ATAM.

Fortalezas del ATAM.

- ✓ Las principales fortalezas en una sesión de evaluación de ATAM son las siguientes:
- ✓ Logro de un mayor entendimiento de la arquitectura por parte de los involucrados.
- ✓ Mejora la documentación arquitectónica y en muchos casos genera documentación nueva.
- ✓ Mejora de la comunicación entre las partes interesadas.

En términos de resultado práctico ATAM ofrece:

- ✓ Calidad de los escenarios producidos por las partes interesadas sobre la base de los atributos de calidad necesarios.
- ✓ Elicitación de los resultados arquitectónicos sobre la base de la calidad de los escenarios y casos de uso.
- ✓ Taxonomías de atributos de calidad, que ofrecen un catálogo con los evaluadores de los parámetros arquitectónicos y los estímulos adecuados para el rastreo de los diferentes atributos de calidad y sus interdependencias.

1.4.2.3 Método de análisis de costo-beneficio. (Cost Benefit Analysis Method, CBAM)

Contexto.

El método CBAM inicia su desarrollo como complemento de ATAM, debido al marcado interés que surge de analizar los costos, beneficios y consecuencias de las decisiones arquitectónicas. Este método además, evalúa el nivel de incertidumbre de estos elementos, que proporciona una base para la toma de decisión con respecto a la arquitectura. Siempre se aplica luego de una sesión evaluativa de ATAM.

Objetivo general.

El objetivo principal de CBAM es tener en cuenta los costos y beneficios como atributos de calidad que contribuye a la sostenibilidad del desarrollo del producto. Así como SAAM y ATAM consideran en el diseño de la arquitectura el cumplimiento de los atributos de calidad arquitectónica como modificabilidad, funcionamiento, disponibilidad, facilidad de uso, entre otros; CBAM reclama la importancia de los

costos, beneficios y riesgos económicos y su valoración a la hora de decisiones arquitectónicas.

Factores claves para su desarrollo.

El impulso del desarrollo CBAM vino dado por un conjunto de preguntas, cada una de las cuales han contribuido en la conformación del método. Estas cuestiones fueron abordadas de la siguiente forma:

- ✓ ¿Cómo pueden las decisiones arquitectónicas medirse y compararse en términos de implicaciones basadas en costos y beneficios?
- ✓ ¿Cómo pueden ser analizados los atributos de calidad, y la interacción entre ellos, en cuanto a los costos y los beneficios que generan su cumplimiento?
- ✓ ¿Cómo se puede caracterizar el nivel de incertidumbre asociado a estas estimaciones de costos y beneficios?

Requisitos previos y entradas.

Como el método CBAM se realiza luego de una sesión de ATAM, esto implica que no serán necesarios algunos requisitos previos como:

- ✓ Una presentación de la arquitectura del sistema, en una notación entendible por todos los involucrados.
- ✓ Familiaridad con conceptos como puntos de sensibilidad, puntos de desventajas, elicitación y descripción de escenarios y requisitos en caso necesarios.

Las entradas en una sesión de evaluación CBAM son las siguientes:

- ✓ Presentación de los objetivos del negocio.
- ✓ Las expectativas del cumplimiento de los atributos de calidad y las limitaciones económicas para lograr esto.
- ✓ La arquitectura y las posibles modificaciones generadas luego de una sesión de evaluación ATAM.

Pasos a realizar para la evaluación.

El método CBAM consta de dos fases. La primera fase es a veces necesaria en caso de que existan un amplio conjunto de propuestas arquitectónicas que se

debatirán, y se hace necesario reducir el número de arquitectura candidatas. Aunque el proceso de evaluación como tal se desarrolla en la segunda fase, conocida con Examen detallado. Para ambas fases se establece en la evaluación CBAM los siguientes pasos:

Paso 1. Elección de los escenarios y las estrategias arquitectónica: En el primer paso se eligen los escenarios que sean de interés la mayoría de los actores del sistema. Para cada uno de ellos se especifica la estrategia de arquitectura a emprender.

Paso 2. Evaluar los beneficios de los atributos de calidad: En esta etapa se obtiene los atributos de calidad relevantes y se valora el beneficio de su cumplimiento, teniendo en cuenta sus implicaciones en el negocio y cómo permita el logro de las funcionalidades del sistema.

Paso 3. Cuantificar los beneficios de las diferentes estrategias de arquitectura: En el primer paso se identificaron las estrategias para cada escenario. Este paso abarca la forma en que una estrategia de arquitectura puede alcanzar el nivel deseado de calidad.

Paso 4. Cuantificar los costos de las estrategias de arquitectura y los plazos: En este paso se obtiene la información de los costos y los calendarios de las partes interesadas, tanto de los gerentes de empresa como los arquitectos. Para lograr esto, se hace necesario que el equipo de evaluación posee suficiente experiencia en la estimación de plazos y costos asociados.

Paso 5. Calcular Conveniencia: Sobre la base de los valores obtenidos en los pasos anteriores, el equipo de evaluación calcula la conveniencia para cada nivel de las estrategias de arquitectura, basándose en la relación beneficio/costo. Además se calcula el nivel de incertidumbre asociado a estos valores, lo que ayuda en la etapa final en la toma de decisiones.

Paso 6. Tomar decisiones: Sobre la base de los resultados en cinco años y el grado de realismo de estos valores, se elige la mejor relación costo-beneficio de las estrategias arquitectónicas, la cual constituye las mejores propuestas para el cumplimiento de los escenarios, y así los atributos de calidad que encierran los mismos.

Funciones de los involucrados.

El método CBAM al igual que los métodos anteriores establece como roles en el proceso de evaluación a los involucrados externos, involucrados internos y equipo de evaluación CBAM. Las diferencias se encuentran en las funciones de cada rol, y las personas que juegan cada rol.

Involucrados externos: Estas personas no tiene participación directa en el proceso de desarrollo de la arquitectura. Ellos tienen como función la presentación de los objetivos del negocio, el sistema de atributos de calidad, así como el nivel de progreso de estos atributos de forma medible, valorando los aspectos económicos. Como ejemplos de involucrados externos se puede mencionar, el equipo de gestión empresarial y el equipo de gestión de proyecto.

Involucrados internos: Este grupo lo constituyen los involucrados que tiene una participación directa en las propuestas de estrategias de arquitectura de software que pueden satisfacer los requisitos de calidad. Tiene como función de analizar, definir y presentar los conceptos de arquitectura para estimar los costos, los calendarios y la incertidumbre asociada en cada una de las estrategias. Ejemplos de involucrados internos son: arquitectos de software y analistas de sistema.

Equipo de evaluación CBAM: El equipo de evaluación no tiene interés directo en el desarrollo de la arquitectura de software del sistema, pero diseña estrategias arquitectónicas en las sesiones de evaluación. Este equipo apoya a los involucrados externos en la presentación de los objetivos del negocio, luego de la presentación de los atributos de calidad relevantes y los escenarios para su obtención. El grupo de evaluación CBAM colabora con los involucrados internos en definir los costos y beneficios de cada una de las estrategias. El equipo consta de jefe de equipo o portavoz, un conjunto de expertos en gerencia empresarial y un secretario.

Esfuerzo necesario para su aplicación.

El equipo de evaluación teniendo en cuenta el tamaño del proyecto y los objetivos del negocio, deben apreciar el esfuerzo necesario. En cuanto a los aspectos organizativos y los pasos para la evaluación CBAM, el esfuerzo se concentra en la definición de las estrategias de arquitectura y análisis de costos, beneficios y calendario de cada una. Un período de sesiones CBAM toma uno o dos días. Como siempre se realiza un período de sesión ATAM, el tiempo total promedia alrededor de cuatro días laborales. En términos de horas-hombre y de

procedimientos de estimación de costos, el equipo CBAM puede proporcionar algunos valores.

Herramientas de apoyo.

No existen aplicaciones de software que permitan la implementación de una sesión de evaluación del método. Las únicas técnicas que pueden ser útiles son métricas y fórmulas de estimación de la incertidumbre que apoyen la realización del paso cuatro, donde se calculan los costos, beneficios y nivel de incertidumbre de cada estrategia arquitectónica.

Alternativas.

Hasta el momento, no hay ningún método que incorpora la perspectiva económica en la evaluación de atributos de calidad de software.

Fortalezas del CBAM.

- ✓ Proporciona una base de valores que permiten realizar una toma de decisiones correcta en la aplicación de determinadas estrategias de arquitectura.
- ✓ Proporciona a la empresa una medida del nivel de rendimiento en la inversión de un cambio en el sistema.
- ✓ Ayuda a las organizaciones en el análisis y pre-evaluación de los recursos a invertir, valorando diferentes alternativas en la adopción de estrategias de arquitectura que permitan la maximización de las ganancias y la reducción de los riesgos.

Comentarios generales.

De manera general, el método CBAM se basa en la evaluación de la arquitectura, al igual que SAAM y ATAM, lo que analiza con énfasis los beneficios económicos de las decisiones arquitectónicas.

1.4.2.4 Método de análisis del nivel de modificación de la arquitectura. (Architecture Level Modifiability Analysis, ALMA)

Contexto.

El método ALMA se ha desarrollado y probado en empresas especializadas en Inteligencia de los Procesos de Negocio (BPI). Este método solo analiza escenarios

basados en la evaluación de los atributos de calidad, centrándose en la modificabilidad del producto. ALMA por sus características, debe ser aplicable en la evaluación arquitectónica de Sistemas Embebidos (ES), pero esta hipótesis no ha sido probada aún.

El análisis de la modificabilidad por lo general abarca los siguientes elementos:

- ✓ Predicción futura de los costos de modificación arquitectónica del producto.
- ✓ Identificación del nivel de flexibilidad del sistema.
- ✓ Comparación de dos o más alternativas de arquitecturas.

Objetivo general.

El método ALMA tiene como objetivo la realización de un análisis correcto basado en la hipótesis de la evaluación de la arquitectura de software a través de la modificabilidad de un conjunto de indicadores: la predicción de los costes de mantenimiento, la evaluación de los riesgos, entre otros. En caso de evaluar y comparar diferentes sistemas, el análisis que realiza el método ALMA apoya el logro de la calidad arquitectónica y la selección de la alternativa correcta. Es por esto que este método utiliza como técnica de evaluación, los escenarios de cambio, proporcionados por las partes interesadas.

El análisis de la modificabilidad que realiza, inicia con la definición de un conjunto de escenarios que pudieran ocurrir durante la evolución del sistema. Estos escenarios se utilizan para verificar la forma en que la actual arquitectura soporta o puede adaptarse a los cambios futuros.

Factores claves para su desarrollo.

El surgimiento del método ALMA fue provocado por un análisis exhaustivo de los principales problemas de los métodos de evaluación arquitectónica existentes. A continuación se plantean las dificultades identificadas:

- ✓ Los métodos anteriores se centran en un único atributo de calidad y hacen caso omiso al resto de los atributos, igualmente importantes.
- ✓ Tienden a realizar análisis muy elaborados y detallados, que requiere en muchas ocasiones gran cantidad de tiempo para lograr un análisis completo.

- ✓ Las técnicas utilizadas por los métodos requieren información detallada que no está disponible durante el diseño de la arquitectura, por ejemplo, la profundidad media de la herencia entre clases, etc.

Otros de los factores que influenciaron la creación del método ALMA fueron:

- ✓ En el ciclo de vida de un sistema de software, el gasto en la evolución del sistema representa del 50 % al 70 % del monto total, por lo que es necesario brindarle atención al nivel de modificabilidad del sistema.
- ✓ Cada día se hace necesario, para mantener una posición en el mercado del software, el logro de mejor calidad, a menos costos y tiempo, lo que el factor modificabilidad del sistema tiene gran influencia.
- ✓ Falta de técnicas de evaluación de arquitecturas de software que cuantifiquen el comportamiento de los atributos de calidad.
- ✓ Falta de métodos que se centren en la modificabilidad del sistema, con la posibilidad de abordar el análisis de múltiples objetivos.

Requisitos previos y entradas.

ALMA se basa en la parte superior de SAAM. Por consiguiente, el método hereda parte de los requisitos previos del SAAM:

- ✓ Las partes interesadas proporcionan los posibles escenarios de cambio que puedan ocurrir en el ciclo de vida futuro del sistema. ALMA utiliza estos escenarios de cambio para analizar la modificabilidad de las arquitecturas.
- ✓ El evaluador debe ser capaz de evaluar el impacto y los costos de estos escenarios de cambio.

Las entradas para una sesión de evaluación ALMA son los siguientes:

- ✓ El "modelo 4 +1", propuesto por Philippe Kruchten para especificar la arquitectura.
- ✓ La notación UML para la descripción de la arquitectura del software.

Pasos a realizar para la evaluación.

El método de evaluación ALMA consta de cinco pasos. Estos pasos no siempre se realizan de forma secuencial. Las iteraciones sobre los pasos también son posibles.

Paso 1. Definir el objetivo de la evaluación: La primera actividad se refiere a la definición del objetivo con que se realizará el proceso evaluativo. El método ALMA puede perseguir distintos objetivos: evaluación de riesgos, estimación del esfuerzo y costo en mantenimiento, o la selección de arquitecturas de software candidatas. La evaluación de riesgos ofrece una visión general sobre los cambios que son difíciles de realizar en una arquitectura determinada. La estimación del esfuerzo y el costo en mantenimiento consiste en valorar los costos que requiere la modificación de la arquitectura seleccionada ante futuros cambios. La selección de arquitecturas candidatas, es la comparación de distintos modelos arquitectónicos propuestos y la selección del más apropiado para el cumplimiento de los atributos de calidad exigidos.

Paso 2. Describir la arquitectura de software: La descripción de la arquitectura se realiza desde una visión arquitectónica, se describe la descomposición de los componentes en el sistema, la relación entre los componentes y las relaciones entre el sistema y su entorno.

Paso 3. Selección de los escenarios de cambios: La selección de los escenarios de cambios es el proceso de búsqueda y selección de las situaciones que pueden desempeñar un papel relevante en el proceso de modificación de la arquitectura de software. Estos tipos de situaciones se utilizan durante la sesión de evaluación ALMA. Obtener escenarios de cambio incluye actividades tales como la identificación de las partes interesadas, entrevistas, la documentación adecuada del resultado que provoca cada escenario de cambio, y evaluar la viabilidad de los resultados, junto con las partes interesadas.

Paso 4. Evaluar el escenario de cambio: Durante esta misión, el analista coopera con los arquitectos y desarrolladores del sistema para determinar el impacto del escenario de cambio y para expresar el resultado de manera adecuada y en concordancia al objetivo que persigue la evaluación.

Paso 5. Interpretar los resultados: Después de evaluar los escenarios de cambio, los resultados se interpretan de conformidad con los objetivos de la evaluación y se verifican contra los requisitos del sistema. Los resultados se utilizan para predecir el

esfuerzo de mantenimiento ante futuros cambios. El valor de la estimación de mantenimiento es limitado y no confiable, ya que no siempre los puntos de referencia u otras estimaciones están disponibles para este atributo.

Funciones de los involucrados.

Dada la participación heterogénea en cada una de las sesiones evaluativas de ALMA las funciones de los involucrados puede agruparse de la siguiente forma:

Interesados externos: Estas personas no tienen participación directa en el proceso de desarrollo de la arquitectura de software. Su función consiste en presentar el contexto empresarial del proyecto, proporcionar los escenarios de cambios y establecer las necesidades iniciales del sistema. En las sesiones finales del proceso evaluativo este grupo es quien decide la continuación o no del desarrollo, basados en los resultados de la evaluación. Ejemplos de involucrados externos son: usuarios, clientes, mantenedores, patrocinadores, propietarios y gerentes del producto que se desarrolla, etc.

Involucrados internos: Este grupo tiene una participación directa en el proceso de desarrollo de la arquitectura de software. Tiene como función analizar, definir y presentar los diferentes conceptos y puntos de vista arquitectónicos. Durante la sesión de evaluación son los responsables de la descripción y la presentación de la arquitectura de software. Junto con el equipo de evaluación son los encargados de estimar el impacto de los escenarios de cambio sobre la arquitectura y evaluar la fiabilidad de los resultados de la evaluación ALMA, según el esfuerzo en lograr la modificabilidad del sistema. Ejemplos de involucrados internos son: arquitectos de software, analistas, diseñadores, desarrolladores, etc.

Equipo ALMA: El equipo de evaluación no tiene participación directa en el diseño de la arquitectura del sistema. Este grupo tiene como objetivo principal la presentación y realización de la evaluación, registro de los artefactos intermedios, evaluar y presentar los resultados finales. Junto con el equipo de arquitectos el equipo ALMA tiene la función de identificar el impacto de las diferentes arquitecturas en los escenarios de cambio y predecir el esfuerzo necesario en la modificabilidad. El equipo ALMA esta compuesto por un líder o portavoz, los analistas de arquitectura de software y un secretario.

Esfuerzo necesario para su aplicación.

Realizando un análisis del número de involucrados, la cantidad de arquitecturas candidatas, así como la cantidad de los escenarios de cambios y la complejidad arquitectónica, son elementos que permiten una estimación del esfuerzo. Sin embargo, en la descripción del método no hay datos reales de estimación de esfuerzo en su aplicación.

Herramientas de apoyo.

El método no cuenta todavía con herramientas de apoyo. En la generación de los escenarios de cambio se utilizan plantillas, reglas o directrices, y manipuladores como pizarras, portafolios, tablas de estimación, notas de reuniones, grabaciones, etc. Para describir la arquitectura de software se utilizan diagramas UML.

Alternativas.

El método ATAM es un posible sustituto de ALMA con respecto a la modificabilidad. Ambos métodos son además, muy similares en el uso de escenarios para la evaluación de los atributos de calidad, y en la estimación del esfuerzo en la realización de modificaciones futuras.

Fortalezas del ALMA.

En general, las fortalezas de una evaluación ALMA son:

ALMA se centra en abstracciones arquitectónicas, que representan la funcionalidad y el dominio de los atributos de calidad.

- ✓ *Los escenarios de evaluación se basan en el análisis de las repercusiones.* Este consiste en identificar los componentes afectados y determinar los efectos sobre los componentes, junto con la onda expansiva que provoca.
- ✓ *Los interesados tienen dos opciones en la generación de los escenarios de cambio.* Un enfoque de arriba hacia abajo, donde de un conjunto de escenarios de cambio, se identifica las categorías, seguido por una mejora en términos de sus casos particulares. Y el enfoque de abajo hacia arriba, donde están los escenarios de cambio recogidos como resultado de las entrevistas con las partes interesadas y, posteriormente clasificados.

- ✓ *La posibilidad de evaluar la modificabilidad desde diferentes perspectivas.*
Estos diferentes enfoques son: el mantenimiento, el costo de predicción, la evaluación de riesgos y la selección de la arquitectura de software.
- ✓ *Proporcionar técnicas que permiten la iteración en los distintos pasos.*

Los resultados del método se pueden resumir en:

- ✓ Los resultados de las estimaciones del impacto de cada escenario se expresan como el tamaño de la modificación a los componentes existentes, o la estimación de tamaño de los componentes que debe ser introducido.
- ✓ Un modelo de predicción basado en las estimaciones de cambio de volumen y ratios de productividad. El modelo asume que el cambio de volumen es el principal generador de costos, lo que da una cifra de productividad por el costo de añadir nuevo elemento y modificar elementos antiguos.

Comentarios generales.

El método, como una observación general, carece de los medios para decidir sobre la fidelidad de los resultados de la evaluación. No permite ALMA predecir la precisión del número de mantenimientos y no brinda seguridad en la integridad de la evaluación de los riesgos.

1.4.2.5 Método de análisis familiar de la arquitectura. (Family – Architecture Analysis Method, FAAM)

Contexto.

El método de evaluación de arquitectura FAAM está enfocado al análisis de grandes sistemas integrados. Se centra en dos atributos de calidad indispensables en el logro de la integración, la interoperabilidad y la extensibilidad.

Objetivo general.

El objetivo del método es el establecimiento de un proceso (con el apoyo de directrices, parámetros y recomendaciones) que evalúe las arquitecturas del sistema de información. A diferencia de otros métodos, FAAM contribuye a:

- ✓ La participación activa de las partes interesadas en el proceso de creación de producto.

- ✓ Centrarse en los atributos de calidad, interoperabilidad y la extensibilidad de los sistemas de información integrados.
- ✓ Haciendo hincapié en los conocimientos prácticos, mecanismos y técnicas que permitan el desarrollo de equipos dentro de la organización para aplicar el método.

Factores claves para su desarrollo.

A continuación se presentan algunos elementos que han contribuido al desarrollo del método FAAM:

- ✓ La necesidad de un método, que ayude a los interesados en la identificación y expresión de los futuros cambios del sistema.
- ✓ La necesidad de técnicas que contribuyen a la interacción entre involucrados y arquitectos en el diseño de la arquitectura.
- ✓ La necesidad de técnicas, que logren una arquitectura integrada en correspondencia con los intereses de las partes interesadas.
- ✓ La necesidad de valorar la integración y la interoperabilidad de los sistemas en las primeras fases de definición de la arquitectura.

Requisitos previos y entradas.

FAAM se basa en los mismos principios generales de evaluación como SAAM y ATAM con especial atención a algunos requisitos previos como:

- ✓ La especificación de la arquitectura debe existir o estar lista antes del período de sesiones FAAM.
- ✓ El facilitador (promotor, arquitecto o las partes interesadas) deben estar dispuestos a expresar lo que se espera de la evaluación.

Las entradas para un período de sesiones de evaluación FAAM son los siguientes:

- ✓ Plantillas que describan los posibles cambios con respecto a la interoperabilidad del sistema y la extensibilidad.
- ✓ Plantillas o técnicas para la generación de mapas que describan las características del sistema integrado, diagramas que contextualicen la integración, los requisitos y criterios para la clasificación.

- ✓ Pueden ser utilizados también directrices o normas para apoyar el proceso de generación de requisitos.
- ✓ La arquitectura se basa en la descripción del "modelo 4 +1" vistas, pero en general, se hace hincapié en la lógica, de proceso y las vistas.

Pasos a realizar para la evaluación.

A continuación se describen cronológicamente los pasos que conforman un período de sesiones FAAM. Es importante señalar que las sesiones FAAM deberán adaptarse en dependencia de la experiencia de la organización en evaluaciones anteriores. El método cuenta con 6 pasos fundamentales.

Paso 1. Definir el objetivo de la evaluación: Este es un ejercicio básico destinado a determinar el objetivo de la evaluación. Con el fin de darle cumplimiento a la actividad se deben abordar aspectos como:

- ✓ Establecer el alcance del sistema integrador por las partes interesadas.
- ✓ Establecer los futuros posibles cambios que afecten la interoperabilidad y la extensibilidad.
- ✓ Proporcionar orientaciones a las partes interesadas para ayudar a generar los requisitos.
- ✓ Proporcionar directrices sobre la priorización de los requisitos para efectuar la evaluación.

Paso 2. Definición de los atributos de calidad a evaluar: En este paso, la participación de los involucrados en la identificación y priorización de los atributos de calidad que debe lograr el sistema, es indispensable. El reto aquí es proporcionar los medios para que los involucrados representen los atributos de forma estructurada, y así queden listos para el proceso evaluativo.

Paso 3. Preparación de la arquitectura a evaluar: En este paso el método se refiere a realizar la representación de la arquitectura en correspondencia con los atributos de calidad establecidos por los involucrados en el paso anterior. El reto aquí es proporcionar directrices a los arquitectos en la representación de los puntos de vistas arquitectónicos.

Paso 4. Revisión de los elementos establecidos para la evaluación: Este paso tiene como objetivo, llegar a un acuerdo sobre el conjunto de las necesidades y

puntos de vista arquitectónicos que son relevantes en el proceso de evaluación. El reto en esta etapa, es aclarar la lógica de negocio o las limitaciones que pueden influir en el avance de la evaluación.

Paso 5. Evaluación de la arquitectura: En este momento, la arquitectura descrita se verifica contra los atributos de calidad especificados, con especial atención en como lograr la interoperabilidad y la extensibilidad ante futuros cambios en la arquitectura.

Paso 6. Elaboración del informe de resultados y propuestas: En esta fase los resultados de la evaluación se registran y se comunica a las partes interesadas. Basándose en los resultados del paso anterior, se proponen soluciones que permitan el cumplimiento de los atributos de calidad y las metas del negocio.

Funciones de los involucrados.

El método FAAM, al igual que los anteriores, adopta la misma clasificación para identificar los grupos que tienen un papel activo durante la evaluación, involucrados internos y externos, y el equipo de evaluación FAAM.

Involucrados internos: Este grupo tiene una participación directa en el proceso de desarrollo de la arquitectura de software. Tiene como función analizar, definir y presentar los diferentes conceptos y puntos de vista arquitectónicos. Durante la sesión de evaluación son los responsables de la descripción y la presentación de la arquitectura de software. Junto con el equipo de evaluación son los encargados de estimar el impacto de los escenarios de cambio sobre la arquitectura y evaluar la fiabilidad de los resultados de la evaluación FAAM, según el esfuerzo en lograr la modificabilidad del sistema. Ejemplos de involucrados internos son: arquitectos de software, analistas, diseñadores, desarrolladores, etc.

Interesados externos: Estas personas no tienen participación directa en el proceso de desarrollo de la arquitectura de software. Su función consiste en presentar el contexto empresarial del proyecto, proporcionar los escenarios de cambios y establecer las necesidades iniciales del sistema. En las sesiones finales del proceso evaluativo este grupo es quien decide la continuación o no del desarrollo, basados en los resultados de la evaluación. Ejemplos de involucrados externos son: usuarios, clientes, mantenedores, patrocinadores, propietarios y gerentes del producto que se desarrolla, etc.

Equipo de evaluación FAAM: El equipo de evaluación no tiene participación directa en el diseño de la arquitectura del sistema. Este grupo tiene como objetivo principal la presentación y realización de la evaluación, registro de los artefactos intermedios, evaluar y presentar los resultados finales. Junto con el equipo de arquitectos el equipo FAAM tiene la función de identificar el impacto de las diferentes arquitecturas en los escenarios de cambio y predecir el esfuerzo necesario en la modificabilidad. El equipo FAAM esta compuesto por un líder o portavoz, los analistas de arquitectura de software y un secretario.

Esfuerzo necesario para su aplicación.

Dado que el esfuerzo en la aplicación de FAAM viene dado por la naturaleza de la evaluación, el nivel de experiencia del equipo y el número de participantes en cada uno de los períodos de sesiones, el intervalo de tiempo necesario, para el desarrollo de una sesión, se puede estimar en un período de 3 días. Estableciendo una comparación con otros métodos y los casos de estudio donde se ha aplicado FAAM, se puede decir que el esfuerzo es relativamente pequeño. La complejidad la brinda siempre el grado de integración del sistema que se evalúa.

Herramientas de apoyo.

FAAM no cuenta actualmente con una herramienta automática que garantice su aplicación. Como se mencionó anteriormente se utilizan Como se indicó anteriormente, este método se apoya en plantillas que describen los posibles cambios con respecto a la interoperabilidad del sistema y la extensibilidad, así como, técnicas para la generación de mapas que describan las características del sistema integrado, diagramas que contextualicen la integración, los requisitos y criterios para la clasificación.

Alternativas.

FAAM se basa en la experiencia de SAAM mediante la adición de una perspectiva de integración y técnicas avanzadas que facilitan la evaluación. La diferencia entre FAAM y ATAM es el alcance de la evaluación. ATAM no se ha aplicado aún a sistemas integrados, pero en principio, puede apoyar. FAAM ha sido diseñado para evaluar la interoperabilidad y la extensibilidad de sistemas integrados. Por lo tanto, el método se apoya en técnicas especializadas en estos atributos, plantillas y directrices consecuentes al proceso evaluativo.

Fortalezas del FAAM.

- ✓ El método proporciona una forma que permite a equipos de desarrollo llevar a cabo su propia evaluación arquitectónica, a través de un proceso de mejora continúa.
- ✓ El proceso de evaluación está diseñado para sistemas integrados, conformado por varias arquitecturas independientes.
- ✓ FAAM tiene un proceso bien definido de trabajo. Esto es útil para apoyar a los participantes con las técnicas de generación de los artefactos necesarios para el proceso de evaluación de los atributos, interoperabilidad y extensibilidad.
- ✓ FAAM está construido sobre la arquitectura general de los métodos de evaluación como SAAM y ATAM, por lo que hereda sus beneficios con respecto a la eficiencia.

A continuación se expone en la tabla 14 una comparación entre los métodos de evaluación antes mencionados.

Métodos	Atributo de calidad	Técnicas de evaluación	Esfuerzo necesario	Tipos de sistema en que es aplicable
SAAM	Modificabilidad	Clasificación de escenarios (directos e indirectos)	Bajo	Todos
ATAM	<ul style="list-style-type: none"> •Modificabilidad. •Seguridad. •Confiabilidad. •Desempeño. 	Escenarios. Selección de los puntos de sensibilidad, desventaja y riesgo.	Medio	Todos
CBAM	Costo/beneficio.	Tiempo y costo.	Alto	Todos
ALMA	Modificabilidad	Escenarios de cambios.	No especificado	Sistema de control embebido, sistemas médicos, telecomunicaciones, sistemas administrativos.
FAAM	Interoperabilidad y extensibilidad.	Tablas y diagramas especializados.	Medio	Sistemas integrados.

Taba 7. Comparación entre los métodos de evaluación SAAM, ATAM, CBAM, ALMA, FAAM.

1.4.2.6 Consideraciones generales sobre los métodos de evaluación de arquitectura.

Los métodos de evaluación constituyen un paso de avance en la mejora del proceso de análisis de la arquitectura de software. Estos brindan herramientas y conceptos que tributan en gran medida al establecimiento de una arquitectura. Los métodos, a diferencia de las técnicas poseen un nivel más alto de exactitud en las formas de medición de los atributos, enfoques y vistas arquitectónicas.

Es el método ATAM, sin duda el más abarcador de toda la colección de métodos examinados. Este no solo utiliza una organización jerárquica en forma de árbol para refinar los requerimientos de la arquitectura, sino que establece métricas normalizadas en valores de máximos y mínimos que dan la medida de los atributos analizados. Tiene como principal desventaja que es un método muy general, difícil de instanciarlo a contextos de sistemas específicos, y que omite el análisis de la portabilidad, factor de calidad importante que influye en la arquitectura de un sistema.

Los métodos de evaluación en su mayoría van enfocados a un único factor de calidad, como es el caso del SAAM y ALMA, con la modificabilidad, y el CBAM con el análisis costo/beneficio. Otra de sus características es que son aplicables en muchas ocasiones a tipos de sistemas determinados, como es el caso de método FAAM en el tratamiento de familias de sistemas.

Estos aspectos traen como consecuencia que sean eficientes en la evaluación de aspectos específicos de una arquitectura, pero que no logren la evaluación de un amplio espectro de arquitecturas de software ni sistemas.

1.5 Experiencia internacional

El estudio de la arquitectura de software desde inicios, ha sido liderado por el Instituto de Ingeniería de Software (Software Engineering Institute, SEI), de la Universidad Carnegie Mellon. Desde la década de 1960 un grupo de especialistas entre los que se pueden citar a Mary Shaw, David Garlan, Paul Clements, Mario Barbacci, Grady Booch, Rick Kazman, entre otros, han puesto todo su talento con el objetivo de desarrollar esta disciplina. (Garlan, y otros, 1994)

El SEI, ha logrado identificar en muchas ocasiones, necesidades que presenta la industria del software para su desarrollo, sobre todo en sistemas de defensa, por estar directamente asociado al Departamento de Defensa de los Estados Unidos.

El tema de evaluar una arquitectura, estimar el comportamiento de los atributos ante cambios arquitectónicos, lograr hacer predicciones de diseño en etapas tempranas, siempre ha sido actividad de vital interés para la industria, debido a que todas estas predicciones aseguran la calidad del software, la disminución de los tiempos de desarrollo, y por lo tanto el esfuerzo y los costos de cada producto. A partir de ello el SEI ha diseñado un conjunto de métodos y técnicas que logran evaluar la arquitectura en etapas tempranas del desarrollo.

Sin embargo, ya hace algunos años, se unió a este conjunto de investigaciones, la Universidad Técnica de Eindhoven, en Holanda, la cual ha tenido un desarrollo vertiginoso en el tema de estimación y evaluación de los atributos de calidad, tanto cualitativa como cuantitativamente. En esta universidad se creó en el año 2001, el grupo de Revisión y Evaluación de Arquitectura de Software (Software Architecture Reviews and Assessment, SARA).

El SARA ha dedicado su trabajo a potenciar la aplicación de métodos de evaluación a través de casos de estudio, y diseñar nuevas herramientas que tributen a esta actividad. (Ionita, y otros, 2006)

Muchas han sido las empresas que han utilizado métodos y técnicas de evaluación para su desarrollo, dentro de ellas se pueden nombrar, control y análisis de tráfico aéreo, servicios de la World Wide Web (www), sistemas para el control de la red de información táctica-estratégica, todos estos dirigidos por el Departamento de Defensa y el SEI. (Mettala, y otros, 1992) y (Clements, y otros, 2005)

Además por la parte de la Universidad de Eindhoven, en dos dominios industriales, la empresa de Consumo Eléctrico (CE) y la empresa de Sistemas Profesionales (PS). Todos los casos con muy buenos resultados en el proceso de evaluación de la arquitectura y la estimación cualitativa y cuantitativa de los atributos de calidad. (Eskenazi, y otros, 2004)

1.6 Conclusiones del capítulo.

En el capítulo se ha presentado, cómo surge el concepto de arquitectura de software, y como ha ido evolucionando hasta nuestros días, para convertirse en una disciplina científica que tiene como objeto de estudio la estructura del sistema, en función de la definición de los componentes y sus interacciones.

Se trata en esta sección, un estudio detallado de dos enfoques importantes a tener en cuenta para la evaluación de la arquitectura de software. Por una parte el análisis desde una perspectiva de calidad, valorando la arquitectura de software como un producto intermedio del ciclo de desarrollo y realizando un análisis de los modelos de calidad enfocados al producto. Aquí se caracterizaron los modelos de McCall, Boehm, Dromey, el modelo FURPS, y el estándar internacional ISO/IEC 9126-1, adecuado para la arquitectura de software.

Desde la otra panorámica, ya más bien orientada hacia aspectos técnicos de la arquitectura de software, aunque sin descuidar la calidad, se analizaron las técnicas y métodos más representativos, entre ellos, las técnicas basadas en escenarios, las basadas en modelos matemáticos, las basadas en simulación, entre otras; y los métodos SAAM, ATAM, CBAM, ALMA y FAAM.

Se realizó además, una valoración de la experiencia internacional en la actividad de evaluación de la arquitectura de software, destacando las principales organizaciones que han dirigido y guiado el quehacer científico entorno a esta actividad. Se mencionan además los resultados de procesos evaluativos en sistemas con entornos de desarrollo diferentes.

Luego de este estudio se concluye la importancia que ocupa la evaluación arquitectónica en el desarrollo de productos de calidad. Permite definir además que el proceso de evaluación de arquitectura no puede ser una actividad que se vea desde un enfoque de calidad arquitectónica únicamente, ni desde un contexto técnico, sino que una evaluación integral que logre todos los beneficios esperados al producto final y al equipo de desarrollo, debe abarcar los dos paradigmas de evaluación, la calidad y los factores técnicos propios de la arquitectura de software.

Capítulo 2: Metodología de evaluación de arquitectura de software.

2.1 Introducción.

En este capítulo se realiza un proceso de caracterización de la actividad productiva de la UCI, el cual sirve de basamento para el diseño de la metodología de evaluación de arquitectura. Se analizan los métodos científicos utilizados en la investigación y con que finalidad se usaron cada uno de ellos. Se fundamenta el método de evaluación técnica que valora la opinión de los expertos en la aceptación de la propuesta o no. Finalmente se realiza la propuesta de la metodología de evaluación arquitectónica con las fases y actividades que la conforman, los involucrados y los artefactos que genera.

2.2 Caracterización de la actividad productiva en la UCI.

Tradicionalmente la universidad genera conocimiento pero es muy difícil que los aplique, por lo que se aleja de los problemas de la producción conllevando a que las investigaciones no respondan a las necesidades de la industria. Por otra parte a la industria le es muy difícil investigar, pues tienen que depender de un mercado cada vez más exigente en tiempo, costo y calidad.

La búsqueda de soluciones ha conllevado a la vinculación universidad-industria; esta es una alianza estratégica de intercambio, donde la primera obtiene la facilidad de aplicar sus investigaciones y de vincular sus estudiantes y profesores al mundo empresarial y la segunda recibe el conocimiento, la innovación constante que generan las universidades, y el empleo de capital humano capacitado, joven y barato en la producción.

Es por ello que las entidades han ido descubriendo que necesitan actualizarse al ritmo de estos avances, de lo contrario estarían condenadas al fracaso; se irían quedando detrás en el vertiginoso impulso del desarrollo. De esta forma, en muchos países, se comenzó a promover la vinculación de los estudiantes y de la universidad en general, con entidades externas. Existen diferentes variantes de esta vinculación, los contratos entre universidades, los parques científicos, y los parques tecnológicos.

La Universidad de las Ciencias Informáticas (UCI) es la primera universidad cubana surgida al calor de la Batalla de Ideas, sobre la base del nuevo concepto de

universidad productiva, logrando una fuerte vinculación estudio-trabajo y universidad-industria. Este nuevo concepto implica que la producción pasa a jugar un papel tan importante en la universidad como la formación. En la UCI los estudiantes y profesores se vinculan a la producción participando en proyectos de alto valor científico y económico, tanto para el mercado nacional, como internacional. Sus resultados contribuyen al desarrollo del país y se plasma la concepción de un fuerte vínculo entre formación-investigación-producción.

La UCI tiene como misión:

- ✓ Formar profesionales, comprometidos con su Patria, altamente calificados en la rama de la informática.
- ✓ Producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

Las facultades que conforman la universidad se vinculan a proyectos productivos y se especializan en segundos perfiles asociados a líneas de producción tales como: Bioinformática, Informática Educativa y Multimedia, Realidad Virtual, Inteligencia Organizacional, Seguridad Informática, Administración de Redes, Software Libre y otras.

El principio del proceso docente educativo es la formación desde la producción, para esto la universidad ha diseñado planes de estudio flexibles. Es generalizado el uso creciente de la teleformación, las teleclases y todos los contenidos de las asignaturas se encuentran en los distintos sitios web, lo que permite el autoaprendizaje. Se acreditan competencias a través de los proyectos productivos y se certifican roles desde el desempeño de cada estudiante en la producción.

El objeto de investigación de la universidad lo constituye el estudio de las TICs. Su principio de diseño es la investigación en función de la producción. En los fundamentos de la política científica de la UCI para el período 2004-2009 se establece: “La UCI continuará avanzando en su conformación como universidad productiva, consolidando el desarrollo científico en función de la producción” (UCI, 2007). La definición de las líneas científicas depende directamente de las necesidades de la producción.

Con el objetivo de guiar metodológicamente el desarrollo de la producción se creó en la universidad la Infraestructura Productiva (IP). La IP presenta un conjunto

de direcciones vinculadas a diferentes líneas de producción y áreas para la gestión proyectos, entre ella se encuentran, la Dirección de Calidad que es la responsable de la liberación de los productos informáticos para su entrega al cliente, y la Dirección Técnica, la cual rige los lineamientos arquitectónicos de la universidad, entornos de desarrollo, arquitectura de información, etc.

Con el objetivo de alcanzar un alto nivel de desarrollo, la UCI como universidad productiva y comercializadora de software, es necesario el logro de un vínculo estrecho entre las actividades de producción, investigación, formación y comercialización. De esta idea surge el concepto de polo productivo, los cuales fueron diseñados como centro de la integración de estas actividades. Los polos productivos constituyen una nueva concepción en la universidad, y se realizan esfuerzos para definir y organizar el establecimiento de estos como célula básica del proceso productivo.

Los polos se organizan y se acometen desde las facultades y se subordinan al Vicedecano de Producción y al Decano. Los jefes de polos son designados por la facultad y la cantera fundamental son profesores. Los equipos de desarrollo están conformados por profesores y estudiantes, que se captan según el año académico que cursan, sus conocimientos y las necesidades de la producción. La definición de la estructura organizacional y la metodología a utilizar es responsabilidad de los líderes de polo, y estos a su vez deben despachar con los Vicedecanos de Producción el avance de cada uno de los proyectos que abarca su polo.

Al convertirse la universidad, en un centro no solamente productor, sino además comercializador de soluciones y servicios informáticos, la calidad en cada uno de los productos que libera la UCI, es un paradigma fundamental en el logro de proyectos de desarrollo exitosos.

2.3 Métodos, procedimientos y técnicas utilizados en la investigación.

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. (Hernández León, 2007) En el presente trabajo se utilizaron como métodos de investigación, los teóricos: histórico lógico, hipotético deductivo, sistémico. Los métodos teóricos permiten comprender el fenómeno que

se estudia, su evolución, elaborar la hipótesis y proponer las mejoras a los problemas que se identifiquen.

Se analizó cómo ha madurado el concepto de arquitectura de software desde su surgimiento hasta la actualidad y el papel destacado que ha tomado la evaluación de las arquitecturas en el mundo industrial.

En la investigación se plantean el problema como un todo, de donde se desprenden los distintos aspectos, según los datos tomados de los proyectos analizados, que reflejan la no realización del proceso de evaluación arquitectónica en una gran gama de proyectos productivos de la universidad.

Se enfocaron las problemáticas de la evaluación de la arquitectura y la verificación de los atributos de calidad en el diseño arquitectónico. En la primera parte del estudio se desarrolló un análisis del estado del arte del enfoque y de la experiencia internacional; se revisaron las bondades y deficiencias de cada uno de los métodos y técnicas en la resolución de esta problemática.

Se analizó además el proceso de evaluación arquitectónica desde el enfoque de la calidad, tomando la arquitectura como producto intermedio del ciclo de desarrollo, y analizando los distintos modelos enfocados a la calidad del producto.

Es importante destacar que la revisión bibliográfica constituyó un método importante para la concepción del proyecto de investigación y para apropiarse de los conocimientos relacionados con el tema.

La investigación fue guiada mediante un método hipotético deductivo porque a partir del problema concreto se plantearon objetivos específicos y la hipótesis, que en el transcurso de la investigación es concretada siguiendo métodos fundamentados.

Unido a la utilización de métodos teóricos, se utilizaron además en el diseño de la investigación los métodos empíricos. Los métodos empíricos permiten describir y explicar las características del fenómeno en estudio. Estos se aplicaron con el objetivo de recolectar los datos necesarios para identificar la problemática y sus causas. Las encuestas y las entrevistas fueron vitales para el diagnóstico de la organización, avalar los conceptos que se manejan en la investigación, medir el alcance y la importancia que tiene la temática. Captar la información cualitativa y cuantitativa del fenómeno, conocer los criterios sobre la evaluación de la

arquitectura de software, así como las posibles soluciones que se proponen en la investigación, fueron otras de las funciones que permitieron el uso de los métodos empíricos.

2.3.1 Entrevista

La población a estudiar fue el personal involucrado en los proyectos de producción de software de la UCI y la unidad de estudio el proceso evaluación de arquitectura en los proyectos productivos. La selección se realizó con la técnica de muestreo no probabilístico, el muestreo intencional, para poder obtener la mayor representatividad e información posible, de acuerdo con los intereses de la investigación. Estos intereses fueron enfocados a entrevistar a personas con experiencia en la definición y seguimiento de la arquitectura de software y que han enfrentado problemáticas que le han permitido ir madurando sus conocimientos sobre la temática.

En el campo de la arquitectura de software, fundamentalmente en la parte de evaluación, la experiencia en la universidad es relativamente baja, algunas especialistas y proyectos están dando sus primeros pasos, por lo que la selección fue basada en este personal que han logrado cierta experticia. Para esto se seleccionaron un conjunto de megaproyectos desarrollados por la universidad que han marcado hitos en el desarrollo productivo de la UCI.

A continuación se reflejan la cantidad de personal entrevistado en cada megaproyecto y el tamaño de la muestra que representan:

Proyecto	Personal entrevistado		
	Arquitectos	A. de calidad	Desarrolladores
<i>Akados</i>	2	1	7
<i>ERP</i>	4	3	14
<i>Registro y notaría</i>	1	2	6
<i>CICPC</i>	2	3	8
<i>GIS</i>	2	2	9
<i>Identidad</i>	2	1	11
<i>Aduana</i>	3	3	8
<i>Alas Pacs</i>	3	2	9
Total	19	17	72
% Muestra	22 %	19 %	14%

Tabla 8. Personal entrevistado por megaproyecto.

Las entrevistas se realizaron con el objetivo de identificar el grado de conocimiento de los involucrados de la situación problemática, validar la propuesta de solución y recopilar elementos a tener en cuenta en la solución. (Ver Anexo 1)

Es importante destacar los resultados arrojados en las entrevistas realizadas al grupo de arquitectura del ERP, debido a sus aportes, conocimientos y experiencias en el proceso de evaluación arquitectónica.

2.3.2 Encuesta

La población a estudiar fue el personal involucrado en el desarrollo de los proyectos de producción de software de la UCI y la unidad de estudio, el proceso de evaluación de arquitectura. La selección se realizó con la técnica de muestreo no probabilística, muestreo intencional, para poder obtener la mayor representatividad e información posible, de acuerdo con los intereses de la investigación que fue la de encuestar a los líderes de proyecto, arquitectos, asesores de calidad y desarrolladores de los proyectos seleccionados. Al igual que en las entrevistas, se tomaron como muestra un conjunto de megaproyecto que desarrollan la universidad en distintas líneas de desarrollo que han marcado la evolución del proceso productivo en la UCI.

Estos megaproyectos abarcan líneas de desarrollo como la gestión empresarial, gestión académica, sistemas georeferenciados, sistemas de gestión médica, entre otras. Esta selección tan heterogénea permitió ampliar el espectro arquitectónico de la evaluación.

Las encuestas fueron orientadas fundamentalmente a la identificación del problema, la valoración de estado de los distintos aspectos que abarca el proceso evaluativo en cada uno de los proyectos. Además se utilizaron para cuantificar de manera representativa, los problemas que se habían identificado, el grado de conocimiento de los involucrados de la situación problemática y del problema, así como su percepción.

En la encuesta se evaluaron los indicadores de la variable del proceso de evaluación arquitectónica, los mismos fueron:

Indicador	Sub-indicador
<i>Atributos de calidad</i>	Definición de los atributos de calidad a evaluar.
	Utilización de modelos de calidad del producto.
<i>Herramientas de evaluación</i>	Uso de métodos de evaluación
	Métodos usa en evaluaciones anteriores

<i>Documentación del proceso</i>	Necesidad de realizar documentación de la evaluación
	Uso de plantillas que estipulen elementos a documentar.

Tabla 9. Indicadores y sub-indicadores de la encuesta aplicada (Ver Anexo 2)

A la hora de elaborar la encuesta se combinaron los tipos de preguntas. La mayoría fueron semicerradas con el interés de conocer la información cuantitativa, pero también de saber la opinión del tema, así como involucrar y motivar a los encuestados en la solución. Se utilizaron además preguntas cerradas, directas e indirectas y de control. (Ver Anexo 2)

2.3.3 Revisión documental.

El enfoque de evaluación de arquitecturas, de estimación del comportamiento de los atributos de calidad en concordancia con cambios arquitectónicos, así como la búsqueda de lograr decisiones de diseño en la fases tempranas del ciclo de desarrollo, son temas muy defendidos en la rama industrial del desarrollo de software, persiguiendo eficiencia y reducción de costos, en un mercado cada vez más exigente de tiempo, costo y calidad.

Es por ellos que se realizó un estudio minucioso de distintos casos de estudios del mundo industrial, donde se reflejan la utilización de técnicas y métodos de evaluación y los resultados que han potenciado. Entre ellos podemos citar, la aplicación del método ATAM, el cual es el más difundido, en la red de información táctica-estratégica (Warfighter Information Network-Tactical, WIN-T), del Departamento de Defensa de los Estados Unidos. Uno de los atributos más tratados fue el rendimiento, debido a la alta exigencia de tiempo respuesta que deben poseer estos sistemas. Los resultados fueron muy positivos, acentuando la relación necesaria establecida entre los involucrados, y la identificación de un conjunto de riesgos. (Clements, y otros, 2005)

Otro de los casos de estudio analizados, fue el análisis cuantitativo de los atributos de calidad de componentes basado en arquitectura, realizado por los holandeses, Evgeny Eskenazi y Alexander Fyukov, de la Universidad Técnica de Eindhoven. En esta investigación se tomaron dos dominios industriales, la empresa de Consumo Eléctrico (CE) y la empresa de Sistemas Profesionales (PS). De las entrevistas realizadas se identificaron dos atributos fundamentales para el logro de

la calidad, el espacio de memoria utilizado, como atributo estático, y el rendimiento, fundamentalmente el tiempo de respuesta del sistema como atributo dinámico. La tendencia en la industria del software es cada vez orientada al desarrollo basado en componentes, y a un amplio nivel de reutilización. Este estudio realizado se basa en este tipo de desarrollo y plantea la imposibilidad de medir cuantitativamente atributos de calidad dinámicos, pues estos no dependen de un único componente, sino de la interacción de muchos. Se propone un método basado en simulación y modelos estadísticos que logra realizar la estimación. (Eskenazi, y otros, 2004)

Este método nos es aplicable a los proyectos de la universidad, por el alto nivel de especificación y de esfuerzo que necesita para su implementación, y el poco conocimiento que existe en nuestros equipos de desarrollo de las técnicas de estimación y evaluación de atributos. Sin embargo, fue muy provechoso el análisis de la investigación, para aclarar conceptos relevantes que se tratan durante el proceso de evaluación de arquitecturas.

Se revisaron además enfoques que miden la evaluación arquitectónica, tomando a la arquitectura como producto intermedio del ciclo de desarrollo, y utilizan un modelo de calidad del producto para establecer el nivel de refinamiento de cada factor de calidad que influye en la arquitectura. De ahí la revisión de artículos científicos que realizan una adecuación del modelo de calidad ISO 9126 (Losavio, y otros, 2003) para el caso específico de la arquitectura de software, donde se presentan casos de estudios representativos que guían el proceso de evaluación. Estos aspectos fueron también de vital importancia en el desarrollo de la investigación.

2.4 Método para la evaluación técnica de la metodología propuesta.

Con el objetivo de realizar la evaluación técnica de la metodología de evaluación de arquitecturas de software propuesta, se utilizó el método de evaluación técnica multi-criterios basado en los aspectos cualitativos evaluados por expertos, documentado en el Curso Básico de Gestión de Proyectos del Dr. Rolando Alfredo Hernández León. El cual permite realizar un estudio de expertos que permita tomar decisiones para aceptar o no el proyecto de acuerdo con los criterios técnicos sobre el mismo. (Hernández León, 2007)

Puede calcularse el número de expertos necesarios, utilizando un método probabilístico y asumiendo una ley de probabilidad binomial mediante la siguiente expresión:

$$n = \frac{p * (1 - p) * k}{i^2}$$

Donde:

- i- Nivel de precisión deseado.
- p- Proporción estimada de errores de los expertos.
- k- Constante asociada al nivel de confianza elegido.

Luego se realiza un proceso de selección de los expertos. La calidad de los expertos influye decisivamente en la exactitud y fiabilidad de los resultados y en ello interviene la calificación técnica, los conocimientos específicos sobre el objeto a evaluar y la posibilidad de decisión entre otros.

En este método se utilizó un conjunto de evaluadores, los cuales poseen cierto grado de experticia en la actividad de definición y evaluación de arquitectura de software, así como la identificación de atributos de calidad necesarios para el aseguramiento de la calidad de un producto en desarrollo. Todos los evaluadores participan en el proceso de producción de software de una manera activa, y para la selección se tuvo en cuenta, los resultados alcanzados en el campo de la arquitectura en productos con un alto nivel de terminación.

Se elaboraron un conjunto de pre-indicadores los cuales fueron sometidos al criterio de los evaluadores, después de recibidas las opiniones y llegado a un consenso se elaboraron los criterios que fueron utilizados para la evaluación. Se definieron cuatro grupos de criterios los cuales son:

- ✓ Criterios de méritos científicos
- ✓ Criterios de implantación
- ✓ Criterios generalización
- ✓ Criterios de impacto

Se le asigna un peso relativo a cada grupo de criterios de acuerdo con las características de la propuesta el cual fue:

- ✓ Grupo No.1..... 20
- ✓ Grupo No.2..... 35
- ✓ Grupo no.3..... 20
- ✓ Grupo No.4.....25

Los evaluadores estudiaron la temática a evaluar, se les entregó la ponencia y dos modelos. Uno guía para que valoraran el peso relativo de cada criterio (Ver Anexo 4.1) y otro para realizar una evaluación cuantitativa de cada criterio con una escala de 1-5 y la apreciación cualitativa con una clasificación final del proyecto en excelente, bueno, aceptable, cuestionable y malo (Ver Anexo 4.2). Con la información recibida dispusieron de un tiempo determinado. También pudieron hacer su valoración final del proyecto, emitiendo todas aquellas consideraciones que estimaron convenientes.

Después, a través del peso relativo otorgado a cada criterio por el trabajo de expertos y la calificación cuantitativa realizada por los evaluadores se elabora un índice de aceptación, lo que unido a la evaluación cualitativa y la evaluación final dada por los evaluadores, permitirá determinar la calidad de la metodología.

Los criterios de evaluación por clasificación fueron:

Grupo No. 1: Criterios de mérito científico

- ✓ Calidad de la investigación.
- ✓ Novedad científica.
- ✓ Valor científico de la propuesta.
- ✓ Aporte científico.

Grupo No. 2: Criterios de implantación.

- ✓ Satisfacción de las necesidades de la producción.
- ✓ Garantía de principios básicos de la arquitectura y la calidad de software.
- ✓ Uso de técnicas, métodos de evaluación de arquitecturas y modelos de calidad.
- ✓ Necesidad del empleo de la metodología.

Grupo No.3: Criterios de generalización

- ✓ Atractividad para su uso.
- ✓ Adaptabilidad a diferentes entornos de desarrollo.

Grupo No.4: Criterios de impacto

- ✓ Repercusión en la calidad del software.
- ✓ Comunicación entre los involucrados.
- ✓ Mitigación de riesgos en etapas tempranas del desarrollo.
- ✓ Ventajas competitivas.
- ✓ Posibilidades de aplicación.

Después de recibir los valores del peso relativo de cada criterio se construye la tabla con los valores del peso relativo de cada criterios, que se muestra en el Anexo 4.3. Luego es necesario verificar la consistencia en el trabajo de expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X_2).

Se sigue el procedimiento siguiente:

Sea **C** el número de criterios que van a evaluarse y **E** el número de expertos que realizan la evaluación. Para cada criterio se determina $\sum E$ que representa la sumatoria del peso dado por cada experto.

Se calcula el peso medio de cada criterio ($M\sum E$) y se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (**S**) por la expresión.

$$S = \sum (\sum E - \sum \sum E / C)^2$$

Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall (W).

$$W = S / E^2 (C^3 - C) / 12$$

El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real.

$$X_2 = E (C-1) W$$

El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas.

Si se cumple:

$$X_{2real} < X_2 (\alpha, c-1)$$

Existe concordancia en el trabajo de expertos. (Hernández León, 2007) Para ello se construye la tabla que se muestra en el Anexo 4.4.

Si existe consistencia en el trabajo de expertos el peso de cada criterio se calcula promediando lo que cada uno de ellos le asignó a cada criterio entre 100, pero si no existe concordancia se hace necesario repetir el trabajo de expertos o eliminar a los expertos que afectan la concordancia.

Conociendo el peso de cada criterio y la calificación dada por los evaluadores en una escala de 1-5 se puede construir la tabla de calificación de cada criterio (Ver Anexo 4.5) y se determina el índice de aceptación del proyecto.

$$IA = P \times C / 5$$

IA: Índice de Aceptación.

P: Peso de los criterios.

C: Criterio promedio concedido por los expertos.

Si:

$IA > 0,7$ Existe alta probabilidad de éxito

$0,7 > IA > 0,5$ Existe probabilidad media de éxito

$0,5 > IA > 0,3$ Probabilidad de éxito baja

$0,3 > IA$ Fracaso seguro

2.5 Metodología de evaluación de arquitecturas de software.

A lo largo de esta investigación se ha venido tratando los motivos de por qué debe ser evaluada una arquitectura de software, los beneficios que trae esta actividad para reducir costos, tiempo y esfuerzo, y a su vez aumentar la calidad de los productos. La UCI como universidad productora, se esfuerza por lograr cada día más un enfoque industrial en sus desarrollos, que no solamente permita insertarnos en mercados de alto renombre internacional, sino que permita aumentar la eficiencia, la calidad y la capacidad de reutilización. La metodología de evaluación de arquitectura propuesta cuenta con dos áreas de procesos fundamentales, el proceso de evaluación (base) y el proceso de documentación. A continuación se exponen las fases y actividades propias de cada proceso.

2.5.1 Proceso de evaluación.

Este proceso está conformado por cuatro fases, concepción, presentación, desarrollo y exposición de resultados, las cuales se desglosan en actividades. Este conjunto de acciones son la que impactan de forma directa sobre la evaluación de la arquitectura de software. Los artefactos que generan cada unas de las actividades son detallados en el proceso de documentación que se efectúa de manera paralela al mismo. A continuación se enumeran las fases del proceso y las actividades de cada una, en orden de sucesión.

Fase de concepción.

1. Selección de los momentos en que se realizará la evaluación arquitectónica a lo largo del ciclo de desarrollo del software.
2. Realización del cronograma de evaluación.
3. Selección de los involucrados.
4. Selección de los integrantes del equipo de evaluación.
5. Preparación de la información.

Fase de presentación.

6. Descripción del desarrollo del proceso evaluativo.
7. Presentación de los objetivos del negocio.
8. Presentación de la arquitectura de software.

Fase de desarrollo.

9. Selección de los atributos de calidad.
10. Refinamiento de los atributos de calidad en indicadores medibles.
11. Evaluación de la arquitectura como producto intermedio del ciclo de desarrollo.
12. Establecimiento de un nivel de prioridad entre los atributos.
13. Identificación de los escenarios asociados a cada atributo de calidad.
14. Descripción de los escenarios identificados.
15. Determinación de puntos críticos de la arquitectura.

Fase de exposición de resultados.

16. Valorar impacto de los posibles cambios arquitectónicos en los atributos de calidad.

17. Realización de cambios arquitectónicos.

18. Exposición de los resultados del proceso de evaluación.

A continuación se exponen detalles de cada una de las actividades, basándose en la definición de los objetivos de cada actividad, participantes, artefactos de entrada y de salida y buenas prácticas identificadas en casos de estudios analizados y consideraciones que propone el autor.

2.5.1.1 Fase de concepción.

Durante esta fase se llevan a cabo todas las actividades de organización, planificación y preparación que aseguran el éxito de las restantes fases.

1. *Selección de los momentos en que se realizará la evaluación arquitectónica a lo largo del ciclo de desarrollo del software.*

Objetivo(s):

Identificar en que fase del desarrollo es propicia evaluar la arquitectura, teniendo en cuenta el objetivo específico de cada evaluación.

Exponer elementos que permitan discernir por parte de los participantes, la relevancia de la evaluación arquitectónica.

Responsable(s):

Grupo de arquitectura.

Participante(s):

Involucrados internos: Líder de proyecto, grupo de arquitectura, analistas del sistema.

Artefacto(s) de entrada:

Artefacto(s) de salida:

Definición de evaluaciones de arquitectura de software: Este artefacto abarca el número de evaluaciones a realizar, momentos en el ciclo de desarrollo que se

realizarán, y una breve descripción del motivo de la evaluación, y la justificación del momento seleccionado.

Buenas prácticas:

El momento de las evaluaciones de arquitecturas es uno de los temas más polémicos en la actualidad. Luego de un estudio realizado en dominios industriales, se propone realizar una evaluación antes de quedar definida la arquitectura de software, que tenga como objetivo evaluar un grupo de propuestas y seleccionar la más indicada. La otra evaluación se propone en el momento que ya queda definida la arquitectura y antes del diseño del sistema, que tendría como objetivo realizar una evaluación del comportamiento de los atributos en correspondencia con la arquitectura establecida y poder tomar decisiones tempranas de diseño, en concordancia con los resultados de la evaluación.

2. Realizar cronograma de evaluación.

Objetivo(s):

Lograr organización y coherencia en las evaluaciones definidas. Realizar cronograma de evaluación que refleje, fecha, lugar, horarios y otros detalles de cada evaluación programada.

Responsable(s):

Líder de proyecto.

Participante(s):

Involucrados internos: Líder de proyecto, grupo de arquitectura.

Artefacto(s) de entrada:

Definición de evaluaciones de arquitectura de software.

Artefacto(s) de salida:

Cronograma de evaluaciones de arquitectura: Este artefacto abarca las evaluaciones definidas, momento en que se realizará, fecha, hora y lugar de cada evaluación. Permite el control del cumplimiento de las evaluaciones a realizar.

Buenas prácticas:

Esta es una actividad novedosa que plantea la metodología propuesta. Se ha diseñado debido a la dinámica de la UCI, y la importancia que tiene para los

proyectos que se desarrollan, una planificación y un control de cada actividad. Unido a esto, como el proceso de evaluación arrastra a un conjunto numeroso de involucrados, la organización, planificación y seguimiento es de vital importancia para lograr reunir a todo el personal necesario. Aquí se propone la utilización de herramientas de gestión de proyectos, como el Dotproject, debido al alto nivel de configurabilidad, no solo establece la planificación de manera global, sino de manera individual, por evaluaciones, etc.

3. Selección de los involucrados.

Objetivo(s):

Selección de los involucrados, externos e internos, que participarán en la evaluación.

Asignar a cada involucrado su misión específica durante la fase de presentación.

Responsable(s):

Líder de proyecto, grupo de arquitectura.

Participante(s):

Involucrados internos: Líder de proyecto, grupo de arquitectura, analistas del sistema.

Involucrados externos: Clientes, usuarios finales.

Artefacto(s) de entrada:

Cronograma de evaluaciones de arquitectura.

Artefacto(s) de salida:

Definición de los involucrados a participar en el proceso de evaluación: Este artefacto abarca las personas que brindarán la información necesaria en la fase de presentación. Tendrá en cuenta datos como, tipos de involucrado (externos o internos), función que realizará durante la fase de presentación y la justificación de su selección para participar en el proceso. Esta fundamentación puede servir de base para futuras selecciones, ya sea en el propio proyecto, o en otros.

Buenas prácticas:

Esta actividad depende directamente del objetivo de la evaluación propia, se debe valorar la participación de la mayoría de los interesados, para el logro de la

consistencia en la información y los análisis que se realicen. Aquí debe hacerse un trabajo diferenciado y de concientización de los involucrados externos con la relevancia del proceso para el desarrollo del sistema, debido a que muchas veces no existe la suficiente comunicación entre ellos y los involucrados internos. El artefacto de entrada ayudará a definir la disponibilidad de los seleccionados a participar en la actividad, en la fecha de evaluación.

4. Selección de los integrantes del equipo de evaluación.

Objetivo(s):

Seleccionar al personal que guiará el proceso de evaluación y apoyará a los involucrados en las distintas definiciones que comprende la fase de desarrollo.

Responsable(s):

Líder de proyecto, involucrados externos.

Participante(s):

Grupo de evaluación: Arquitectos con experiencia en proceso de evaluaciones anteriores, asesores de calidad.

Artefacto(s) de entrada:

Cronograma de evaluaciones de arquitectura.

Artefacto(s) de salida:

Definición del grupo de evaluación: Este artefacto abarca las personas que brindarán contarán con la experticia en las distintas actividades del proceso evaluativo. Tendría en cuenta datos como, cantidad de evaluaciones realizadas, resultados relevantes de evaluaciones anteriores formando parte del equipo evaluador. Estos elementos pueden servir de base para futuras selecciones, ya sea en el propio proyecto, o en otros. El artefacto de entrada ayudará a definir la disponibilidad de los seleccionados en la fecha de evaluación.

Buenas prácticas:

El equipo evaluador tiene como principal desventaja, el desconocimiento de las metas del negocio, las características del equipo de desarrollo, entre otras informaciones propias del proyecto. Se propone antes iniciar la evaluación brindarle documentación que los ayude a comprender las propiedades distintivas del proyecto.

5. Preparación de la información.

Objetivo(s):

Diseñar todas las presentaciones que tributen información al análisis arquitectónico, la selección de atributos de calidad esperados y la definición de los escenarios, a exponer en la fase de presentación.

Responsable(s):

Involucrados externos e internos seleccionados en la actividad 3.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas y asesores de calidad.

Artefacto(s) de entrada:

Definición de evaluaciones de arquitectura de software.

Artefacto(s) de salida:

Definición de las metas del negocio: Este artefacto abarca elementos como, arquitectura de información, descripción de los procesos que debe soportar el sistema con sus flujos alternativos, actores del negocio, etc. Todos estos elementos deben ser presentados con una panorámica arquitectónica, o sea de que forma impactan sobre el diseño y arquitectónico y viceversa.

Descripción de la(s) arquitectura(s): Descripción detallada de los componentes que conforman la arquitectura, las interacciones entre ellos, y las restricciones y riesgos de cada decisión arquitectónica. La forma de presentación debe ser orientada más bien, a como la arquitectura da cumplimiento a las metas del negocio propuestas en el artefacto anterior.

Buenas prácticas:

Esta actividad debe lograr un alto grado de especificación y preparación, para que las exposiciones permitan promover valoraciones útiles para el análisis y no se desvanezcan en detalles insignificantes. Se deben realizar un conjunto de revisiones, de manera periódica que logre la calidad y la precisión en las exposiciones. El artefacto de entrada brindará el objetivo de la evaluación a realizar.

2.5.1.2 Fase de presentación.

Durante esta fase se exponen toda la información necesaria para el proceso evaluativo, metas del negocio, arquitectura de información, arquitectura de software, valoraciones del equipo de evaluación, etc.

6. Descripción del desarrollo del proceso evaluativo.

Objetivo(s):

Describir de que forma se llevará a cabo el proceso de evaluación, número de sesiones, tiempo estimado de cada una, temas por sesiones, momento de exposición de los resultados, y responsables.

Responsable(s):

Equipo de evaluación seleccionado en la actividad 4.

Participante(s):

Arquitectos con experiencia en proceso de evaluaciones anteriores, asesores de calidad.

Artefacto(s) de entrada:

Definición de evaluaciones de arquitectura de software.

Cronograma de evaluaciones de arquitectura.

Artefacto(s) de salida:

Cronograma de sesiones de la evaluación: Este artefacto abarca elementos como, número de sesiones estimadas a realizar, participantes por sesión, responsables, fecha, hora y lugar de cada sesión, tiempo estimado, objetivo de cada sesión, resultados esperados al concluir y las actividades que abarca cada sesión.

Buenas prácticas:

En esta actividad se introduce nuevamente los factores organización, planificación y seguimiento de cada una de las sesiones, con el objetivo principal de lograr la participación de todas las personas necesarias en las sesiones del proceso evaluativo y el éxito en cada una de ellas. Aquí se propone la utilización de herramientas de gestión de proyectos, como el Dotproject, debido al alto nivel de configurabilidad, no solo establece la planificación de manera global, sino de manera individual, por sesiones, etc.

7. Presentación de los objetivos del negocio.

Objetivo(s):

Presentar toda la información preparada por los involucrados externos en la actividad 5.

Responsable(s):

Involucrados externos.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas y asesores de calidad y equipos de evaluación.

Artefacto(s) de entrada:

Definición de las metas del negocio

Artefacto(s) de salida:

Valoraciones de los objetivos del negocio: Este artefacto tiene como objetivo registrar los elementos relevantes que aportó el debate, de todos los participantes en esta actividad, que sirva de referencia a actividades posteriores.

Buenas prácticas:

Lograr la claridad y la precisión de estas exposiciones es de vital importancia para el proceso de evaluación. Promover discusiones, puntos de vistas diferentes, aclarar dudas, son actividades que enriquecen la evaluación arquitectónica, y permite un alto grado de análisis por parte de todos los participantes.

8. Presentación de la arquitectura de software.

Objetivo(s):

Realizar una presentación de la arquitectura establecida o el conjunto de arquitecturas candidatas, en dependencia del objetivo y momento de la evaluación.

Responsable(s):

Grupo de arquitectura.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas y asesores de calidad y equipos de evaluación.

Artefacto(s) de entrada:

Descripción de la arquitectura.

Artefacto(s) de salida:

Valoraciones de la arquitectura presentada: Este artefacto tiene como objetivo registrar los elementos relevantes que aportó el debate, de todos los participantes en esta actividad, que sirva de referencia a actividades posteriores.

Buenas prácticas:

Eliminar términos y palabras técnicas que eviten el entendimiento con claridad por los participantes, de todos los elementos que componen la arquitectura. Guiar la presentación de manera que se exprese con exactitud cómo la composición arquitectónica da cumplimiento a las metas del negocio, y de ser posible valorar el comportamiento de algunos atributos de calidad.

2.5.1.3 Fase de desarrollo.

Durante el transcurso de esta fase es que se realiza el proceso evaluativo propiamente.

9. Selección de los factores de calidad.

Objetivo(s):

Seleccionar los factores de calidad, que para su validación influya directamente la arquitectura del sistema.

Responsable(s):

Líder de proyecto.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas, asesores de calidad y equipo de evaluación.

Artefacto(s) de entrada:

- ✓ Valoraciones de los objetivos del negocio.
- ✓ Valoraciones de la arquitectura presentada.
- ✓ Adecuación del modelo de calidad ISO 9126 para la arquitectura de software.

Artefacto(s) de salida:

Definición de los factores de calidad: Este artefacto abarca todas las características de calidad necesarias para el logro de la calidad de la arquitectura como producto intermedio del ciclo de desarrollo.

Buenas prácticas:

Para la selección de los factores de calidad, podemos utilizar algunos de los modelos existentes. Se propone el uso de la adaptación del modelo ISO 9126 de calidad de software para efectos de la evaluación de arquitecturas de software, propuesto por Losavio y otros autores. El modelo se basa en las características que se relacionan directamente con la arquitectura: funcionalidad, confiabilidad, eficiencia, mantenibilidad y portabilidad. Si fuera necesaria la adición de un nuevo factor de calidad, además de los mencionados anteriormente que establecen una relación directa con el diseño arquitectónico, que se lleve a consideración de todos los participantes en el proceso de evaluación, previa justificación.

10. Refinamiento de los atributos de calidad en indicadores medibles.

Objetivo(s):

Llevar a cabo un refinamiento de los factores de calidad establecidos en la actividad anterior, hasta llegar a lograr la verificación o validación de cada atributo de calidad mediante métricas o indicadores medibles.

Responsable(s):

Equipo de arquitectura.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas, asesores de calidad y equipos de evaluación.

Artefacto(s) de entrada:

Definición de los factores de calidad.

Artefacto(s) de salida:

Lista de chequeo de los atributos de calidad: Este artefacto consiste en una lista de chequeo de cada uno de los factores refinados en atributos de calidad, asociados a métricas o indicadores medibles.

Buenas prácticas:

Esta actividad es la que materializa el diseño de la evaluación de la arquitectura de software desde el enfoque de la calidad arquitectónica. Es de vital importancia que todos los factores sean refinados en atributos de calidad que puedan ser medidos mediante métricas o indicadores. Para esto se propone el uso de la adecuación del modelo ISO 9126 para efectos de la arquitectura de software.

Este modelo realiza un refinamiento de cada una de las características que propone para la evaluación hasta un nivel arquitectónico, reflejando indicadores medibles que verifican como la arquitectura propuesta cumple con los factores de calidad de un producto. Todos los involucrados en el proceso de evaluación pueden participar en la propuesta de nuevas métricas e indicadores.

11. Evaluación de la arquitectura como producto intermedio del ciclo de desarrollo

Objetivo(s):

Llevar a cabo el proceso de medición de cada atributo de calidad según la lista de chequeo establecida en la actividad anterior. Brindar conclusiones y resultados sobre los valores obtenidos en cada atributo analizado.

Responsable(s):

Líder de proyecto.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas, asesores de calidad y equipos de evaluación.

Artefacto(s) de entrada:

Lista de chequeo de los atributos de calidad.

Artefacto(s) de salida:

Valores resultantes de cada atributo. Este artefacto abarca los resultados de la aplicación de la lista de chequeo establecida en el paso anterior. Debe reflejar factor de calidad, criterios asociados, atributo de calidad, métrica o indicador y resultado.

Comentarios y resultados finales. Este artefacto muestra, basándose en los valores resultantes de cada atributo, conclusiones valorativas sobre el cumplimiento de la calidad en la arquitectura de software.

Buenas prácticas:

Esta actividad constituye el cierre del proceso evaluativo de la calidad arquitectónica. Es muy importante que los comentarios y resultados finales sean discutidos por los participantes en el proceso de evaluación, y se emitan criterios valorativos sobre los resultados. Puede darse el caso ante arquitecturas candidatas que estos resultados no puedan ser definitorios, debido a la igualdad de atributos cumplidos, sin embargo, las siguientes actividades de la fase de desarrollo contribuirán a enriquecer estos resultados con el fin de que se pueda tomar una decisión acertada.

12. Identificación de los escenarios.

Objetivo(s):

Identificar los escenarios que permita evaluar el comportamiento de los atributos definidos.

Responsable(s):

Equipo de arquitectura.

Participante(s):

Clientes, usuarios finales, administradores del sistema, líder de proyecto, arquitectos, analistas y asesores de calidad y equipos de evaluación.

Artefacto(s) de entrada:

Definición de los atributos de calidad.

Artefacto(s) de salida:

Identificación de los escenarios: Este artefacto abarca el conjunto de escenarios que permiten evaluar el comportamiento de los atributos de calidad definidos. Se expondrá una breve descripción de los escenarios y atributo o atributos asociados al mismo.

Buenas prácticas:

La identificación de los escenarios depende en gran medida de la comprensión del comportamiento de los atributos de calidad, y de las interacciones que establecen cada uno de los actores del sistema, los componentes arquitectónicos que intervienen, entre otros elementos. Es muy útil lograr en esta actividad, la sinergia entre involucrados y equipo de evaluación, unos pues son lo que conocen el funcionamiento del sistema y requisitos que debe cumplir el mismo, y otros porque poseen la experiencia obtenida en evaluaciones anteriores.

13. Descripción de los escenarios.

Objetivo(s):

Realizar una descripción detallada de todos los elementos que conforman cada uno de los escenarios identificados.

Responsable(s):

Equipo de arquitectura.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, líder de proyecto, analistas y asesores de calidad y equipos de evaluación.

Artefacto(s) de entrada:

Identificación de los escenarios

Artefacto(s) de salida:

Descripción detallada de los escenarios: Este artefacto abarca la descripción detallada de cada escenarios definido, teniendo en cuenta, aspectos como: origen del estímulo, estímulo, ambiente, componente, respuesta y medida de la respuesta.

Buenas prácticas:

Para identificar cada uno de los aspectos a tener en cuenta en la descripción detallada de los escenarios es importante conocer:

- ✓ El origen del estímulo es la acción de cualquier actor que interactúa con el sistema.
- ✓ El estímulo es una condición que necesita ser considerada cuando se produce la acción del actor.

- ✓ Ambiente son las condiciones en la cual se encuentra el sistema en el momento en que recibe el estímulo.
- ✓ Los componentes del sistema son aquellos afectados al producirse el estímulo.
- ✓ La respuesta es la actividad que debe realizar el sistema.
- ✓ Y la medida de la respuesta es un tiempo de medida con el cual debe cumplir la respuesta para que el requerimiento pueda ser testeado.

Para esta descripción detallada es necesario la participación y unión de los criterios de los involucrados y el equipo de evaluación.

14. Establecimiento de la prioridad de los escenarios.

Objetivo(s):

Establecer la prioridad de cada uno de los escenarios definidos.

Responsable(s):

Equipo de arquitectura.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas y asesores de calidad y equipos de evaluación.

Artefacto(s) de entrada:

- ✓ Identificación de los escenarios
- ✓ Descripción detallada de los escenarios

Artefacto(s) de salida:

Grado de prioridad de los escenarios: Este artefacto abarca la organización de los escenarios según su relevancia para el logro de la calidad.

Buenas prácticas:

La prioridad de los escenarios se debe establecer teniendo en cuenta los atributos que encierra y la prioridad de cada uno de ellos. En caso de que el escenario implique a más de un atributo, es importante observar la existencia de conflictos entre los atributos involucrados.

15. Determinar puntos críticos.

Objetivo(s):

Determinar puntos de sensibilidad, puntos de riesgos y puntos de desventajas de la arquitectura diseñada.

Responsable(s):

Equipo de arquitectura.

Participante(s):

Clientes, usuarios finales, administradores del sistema, arquitectos, analistas y asesores de calidad, líder de proyecto y equipos de evaluación.

Artefacto(s) de entrada:

- ✓ Definición de los atributos de calidad.
- ✓ Identificación de los escenarios.
- ✓ Descripción detallada de los escenarios.
- ✓ Grado de prioridad de los escenarios.
- ✓ Requisitos del sistema.
- ✓ Descripción de la arquitectura.

Artefacto(s) de salida:

Puntos críticos de la arquitectura: Este artefacto abarca los puntos de sensibilidad, los puntos de riesgos y de desventaja de la arquitectura diseñada. Refleja además, una breve descripción de cada punto y los atributos involucrados.

Buenas prácticas:

Para identificar todos los elementos de esta actividad es necesario tener en cuenta:

- ✓ Un punto de sensibilidad es la colección de componentes de la arquitectura que son fundamentales para el logro de un atributo de calidad en particular.
- ✓ Los puntos de desventajas es un punto de sensibilidad que es fundamental para el logro de múltiples atributos.

- ✓ Los puntos de riesgos son un subconjunto de los puntos de sensibilidad que puede evitar el logro de las metas de calidad del atributo.

2.5.1.4 Fase de exposición de los resultados.

Durante esta fase se valoran las posibles modificaciones, se efectúan los cambios pertinentes, se concluye el proceso de evaluación y se redactan las memorias del proceso evaluativo, como documento de cierre del proceso.

16. Valorar impacto de los posibles cambios arquitectónicos en los atributos de calidad.

Objetivo(s):

Identificar posibles cambios o modificaciones que se puede aplicar a la arquitectura.

Valorar el impacto de estos cambios sobre el comportamiento de los atributos de calidad.

Responsable(s):

Equipo de arquitectura.

Participante(s):

Arquitectos, analistas y asesores de calidad, líder de proyecto y equipos de evaluación.

Artefacto(s) de entrada:

- ✓ Definición de los atributos de calidad.
- ✓ Lista de chequeo de los atributos de calidad.
- ✓ Puntos críticos de la arquitectura.

Artefacto(s) de salida:

Identificación de posibles cambios: Este artefactos abarca el conjunto de cambios propuestos luego de análisis llevado a cabo en actividades posteriores. Entre los elementos que se registre de cada propuesta, especificar el impacto que causaría el cambio sobre el comportamiento de los atributos de calidad que involucra.

Buenas prácticas:

Hacer énfasis en el comportamiento de los atributos ante los cambios, la prioridad de cada uno de ellos frente al logro de la calidad, y el grado de conflicto de estos atributos que se puede establecer ante una modificación arquitectónica.

17. Realización de cambios arquitectónicos.

Objetivo(s):

Esta actividad abarca dos objetivos, en dependencia del momento de la evaluación.

En caso de una evaluación de arquitectura candidatas, esta actividad tiene como función elegir la arquitectura más adecuada.

De ser una evaluación que se centra en el grado de cumplimiento de los atributos de calidad, dado un diseño arquitectónico establecido, en este momento se establecen las modificaciones arquitectónicas a llevar a cabo. Siempre tomando como premisa el análisis realizado en las actividades anteriores.

Responsable(s):

Equipo de arquitectura.

Participante(s):

Arquitectos, analistas y asesores de calidad, líder de proyecto y equipos de evaluación.

Artefacto(s) de entrada:

Identificación de posibles cambios

Artefacto(s) de salida:

Descripción de la arquitectura: En esta actividad, en caso de haber cambios en la arquitectura del sistema, se actualiza el artefacto de Descripción de la arquitectura establecido en la actividad 5.

Buenas prácticas:

Durante esta actividad es necesario un alto grado de análisis que permita tomar decisiones acertadas, que permitan los resultados esperados en etapas posteriores del ciclo de desarrollo.

18. Exposición de los resultados del proceso de evaluación.

Objetivo(s):

Realizar las conclusiones del proceso evaluativo.

Exponer los resultados obtenidos del proceso de evaluación de arquitectura de software.

Responsable(s):

Equipo de evaluación

Participante(s):

Involucrados externos e internos y equipo de evaluación.

Artefacto(s) de entrada:

Descripción de la arquitectura (versión final).

Artefacto(s) de salida:

Memorias del proceso de evaluación: Este artefacto abarca un resumen de todos los aspectos relevantes del proceso, atributos, breve descripción de su comportamiento y prioridad, los escenarios y una breve descripción de cada uno.

2.5.2 Proceso de documentación.

El proceso de evaluación es de suma importancia para la metodología propuesta, pues es quien permite la reutilización de toda la información que genera cada una de las evaluaciones, y permite crear una base de conocimientos que asegure la calidad de los procesos evaluativos en presencia de personal experto o no.

Para esto se definieron un conjunto de artefactos documentales que se enumeran a continuación:

- ✓ Definición de evaluaciones de arquitectura de software
- ✓ Cronograma de evaluaciones de arquitectura
- ✓ Definición de los involucrados a participar en el proceso de evaluación
- ✓ Definición del grupo de evaluación
- ✓ Descripción de las metas del negocio
- ✓ Descripción de la(s) arquitectura(s)

- ✓ Cronograma de sesiones de la evaluación
- ✓ Valoraciones de los objetivos del negocio
- ✓ Valoraciones de la arquitectura presentada
- ✓ Listado de los factores de calidad
- ✓ Lista de chequeo de los atributos de calidad
- ✓ Valores resultantes de cada atributo
- ✓ Comentarios y resultados finales de la calidad arquitectónica.
- ✓ Identificación de los escenarios
- ✓ Descripción detallada de los escenarios
- ✓ Grado de prioridad de los escenarios
- ✓ Puntos críticos de la arquitectura
- ✓ Identificación de posibles cambios
- ✓ Memorias del proceso de evaluación

A razón de normar los elementos que debe tener cada uno de los artefactos y lograr un proceso de documentación detallado y eficiente, que permita el cumplimiento del objetivo general de este proceso, la reutilización de las experiencias en los procesos evaluativos en proyectos de desarrollo similares, se diseñaron un conjunto de plantillas que conforman el expediente de la evaluación de la arquitectura, el cual es presentando en el Anexo 5.

2.6 Conclusiones del capítulo.

En este capítulo se expusieron las características distintivas que posee la UCI como universidad productiva y la necesidad de guiar sus desarrollos cada día más hacia una tendencia industrial. Se analizaron los métodos científicos utilizados en la investigación, las entrevistas, encuestas y revisiones documentales realizadas. Y por último se presentó la metodología de evaluación de arquitectura propuesta por esta investigación. Se expusieron los procesos que encierra la metodología, así como cada una de las fases, actividades y artefactos que abarca.

Capítulo 3. Proceso de implementación de la metodología.

3.1 Introducción.

En este capítulo describe los pasos a seguir para lograr una aplicación exitosa del uso de la metodología de evaluación de arquitectura de software en los proyectos productivos de la universidad. Realiza un análisis de los resultados y las encuestas aplicadas, con el objetivo de establecer un nivel de partida que contribuya a la comparación con los resultados de la aplicación. Analiza los resultados de la evaluación técnica de la propuesta brindando una panorámica de la opinión de los expertos y finalmente realiza un análisis valorativo de los resultados del uso de la metodología en un proyecto piloto.

3.2 Proceso de implementación de la metodología.

En el capítulo anterior de esta tesis, se expuso el diseño de una metodología de evaluación de arquitectura de software. En ella se definen las fases que componen la metodología, las actividades que abarca cada una de las fases, incluyendo responsables, participantes, artefactos de entrada y salida, así como un conjunto de buenas prácticas y consideraciones a tener en cuenta por cada una de las actividades definidas.

La metodología de evaluación fue elaborada para su aplicación en cualquiera de los proyectos que desarrolla la universidad, a pesar de que las líneas de desarrollo sean heterogéneas. Solamente deben estar creadas las condiciones que propicien un proceso evaluativo con calidad que permita el cumplimiento de todos los objetivos propuestos en cada una de las evaluaciones.

Para esto se diseñaron un conjunto de acciones a cumplir que brinden apoyo a la aplicación exitosa de la metodología en cada una de sus fases. Estas acciones vienen especificadas por un conjunto de normas o condiciones que deben estar garantizadas, antes de la evaluación, durante y al finalizar la misma. A continuación se explican en detalle:

3.2.1 Acciones que garantizan el nivel de partida de la evaluación.

Lograr que todo el personal del proyecto productivo conozca la importancia del proceso de evaluación de arquitectura. En la fase de concepción de la metodología se seleccionan los momentos en que durante el ciclo de desarrollo se realizarán las evaluaciones arquitectónicas, se selecciona el personal a participar en la evaluación y se conforma el cronograma de ejecución de la evaluación. Antes de iniciar este proceso organizativo es necesario que el arquitecto principal del proyecto, o un integrante del equipo de arquitectura hagan una exposición acerca de los beneficios de la evaluación. Este paso es útil a la hora de lograr compromiso de cada participante en las sesiones de evaluación, en muchas ocasiones difícil de alcanzar debido a la dinámica de la universidad.

Lograr que todo el personal involucrado en la evaluación conozca la metodología que usarán durante el proceso evaluativo. Durante la fase de concepción se selecciona las personas que van a participar en la evaluación. Se hace imprescindible una breve exposición de la metodología, y el desempeño esperado de cada participante en el logro de un proceso satisfactorio. Esta acción es recomendable que la realice el líder del proyecto, y de ser posible ya asocie responsables a cada actividad que conforma la metodología.

Lograr que los participantes en la evaluación estén identificados con la documentación a usar. Como se explica en el capítulo anterior de este trabajo, el proceso de documentación genera un conjunto de artefactos. Para mayor entendimiento de los involucrados se diseñaron un conjunto de plantillas que permiten reflejar uniformemente todos los elementos que conforman un artefacto generado durante el proceso de evaluación. Es necesario que estas plantillas sean expuestas y discutidas con todas las personas que conforman el equipo, a razón de que no exista incongruencias en su confección, y evitar posibles atrasos en la elaboración de cada artefacto.

3.2.2 Acciones que facilitan el proceso de evaluación y garantizan la calidad del mismo.

Garantizar que todos los involucrados en la evaluación tengan conocimientos avanzados de arquitectura de software. Esta acción constituye más bien una norma preestablecida a la selección del equipo de evaluación. Sin embargo, la realidad

demuestra que es un requisito de un alto nivel, en la mayoría de los casos, imposible de cumplir en todos los que conforman el equipo de evaluación, es por ello que se propone como acción. Aquí se recomienda realizar una breve capacitación al equipo de evaluación, orientada más bien al entendimiento de las arquitecturas evaluadas y una justificación teórica de su diseño. Esta labor contribuirá en gran medida a la valoración de cambios arquitectónicos.

Garantizar que todos los involucrados en el proceso evaluativo cuenten con conocimientos de modelos de calidad del producto y refinamiento de los atributos en indicadores medibles. Estos conceptos deben ser dominados fundamentalmente por los asesores de calidad que conforman el equipo de evaluación. Sin embargo, es necesario debido al desconocimiento de muchos arquitectos y desarrolladores de estas temáticas, que se lleve a cabo una exposición de la adecuación del modelo de calidad ISO 9126 para efectos de la arquitectura de software y se analice un caso de estudio. Para esto se puede tomar como referencia este documento.

Garantizar una buena planificación de las sesiones de evaluación. Es necesario que la planificación del tiempo de las sesiones de evaluación se realice de manera mesurada. Cada uno de los encuentros evaluativos deben estar normados por un tiempo, con el objetivo que los primeros encuentros no se alarguen en el tiempo, y luego los venideros, se realicen de manera atropellada. Debe ser responsabilidad del líder de proyecto que se cumplan los tiempos establecidos para cada actividad, para que no provoca agotamiento en los participantes y disminuya la calidad del proceso.

Garantizar que el proceso de documentación se realice paralelo al proceso de evaluación. Es indiscutible que el proceso de documentación en muchas ocasiones resulta tedioso para los participantes de un proyecto productivo en la universidad. Sin embargo, el proceso de documentación es el que permite no solo replicar experiencias en proyectos similares, sino que orienta al equipo en el nivel de partida de cada actividad, luego de un análisis del artefacto de entrada. Además, el artefacto de salida establece de manera concisa los elementos resultantes a tener en cuenta para un exitoso desarrollo de la actividad. Es por ello que se requiere no dejar para el final el proceso de documentación e ir a la par del proceso de evaluación.

3.2.3 Acciones que garantizan replicar experiencias en proyecto similares.

Crear una base de conocimientos. Se recomienda crear a nivel de polo o área temática una base de conocimientos con todos los expedientes de cada unas de las evaluaciones realizadas. Este material es de utilidad para el propio proyecto, como base de futuras evaluaciones y para el resto de los proyectos que conforman el polo o área temática por la similitud de sus arquitecturas. Se propone la utilización del sistema de gestión documental Alfresco.

Estas acciones recomendadas facilitarán el desarrollo del proceso de evaluación de arquitectura de software, contribuirá a elevar el nivel de la calidad del proceso, influyendo de manera directa en la calidad del producto final. Permitirá auxiliar a los equipos que no cuenten con experticia y experiencia en procesos de evaluación arquitectónica, que constituyen un conjunto considerable entre los proyectos productivos de la universidad.

3.3 Análisis de los resultados de la aplicación de las encuestas y entrevistas.

Se entrevistaron un total de 10 personas, especialistas del grupo de arquitectura de la Dirección Técnica, especialistas del grupo de normalización de la Dirección de Calidad, integrantes del grupo de arquitectura del ERP y arquitectos del polo de Petrosoft y de Video y Sonido Digital de la facultad 9.

Las entrevistas manifestaron lo necesaria que sería la realización de la evaluación arquitectónica en diferentes momentos del ciclo de desarrollo, en especial en las etapas iniciales. La mayoría de los entrevistados coinciden en que en la mayoría de los proyectos no se lleva a cabo una valoración arquitectónica y esto trae consecuencias fatales durante la etapa de implementación, lo que provoca grandes retrasos en los desarrollos.

Sin embargo se observa en casi todos los entrevistados un desconocimiento acerca de que es en sí una evaluación arquitectónica, y que competencias debe poseer el equipo de evaluación para llevar a cabo el proceso evaluativo de manera exitosa. Las entrevistas evidenciaron además, el conocimiento de algunos métodos de evaluación de arquitectura, en especial el método ATAM, pero en contraste a

esto, un gran número de entrevistados plantean que no utilizan ninguna técnica ni método a la hora de evaluar la arquitectura de su proyecto.

Cabe destacar los resultados de las entrevistas al grupo de arquitectura del ERP, los cuales poseen un gran dominio de todas las técnicas y métodos de evaluación, y plantean que se encuentran inmersos en una investigación que dará como resultado la creación de un método de evaluación propio.

De manera general todos los entrevistados coinciden que un proceso de evaluación de la arquitectura de software diseñada en sus proyectos, traería ventajas como:

- ✓ Mejorar las decisiones arquitectónicas.
- ✓ Verificar el cumplimiento de los atributos de calidad en la arquitectura del sistema.
- ✓ Tomar un conjunto de decisiones de diseño en etapas tempranas del desarrollo.
- ✓ Reducir riesgos en la fase posteriores al diseño arquitectónico.

Sin embargo, es generalizada la opinión de la falta de conocimientos de las herramientas, técnicas, modelos de calidad y métodos que le permiten llevar a cabo un proceso evaluativo.

Con vistas a cuantificar la información recogida, se encuestaron 19 arquitectos, 17 asesores de calidad y 72 desarrolladores, a un total 108 personas de una muestra de 8 megaproyectos de la universidad, ERP, Akademos, CICPC, Alas Pacs, Identidad, Registro y Notaría, Aduana y GIS. En la figura # se muestra el gráfico de pastel que representa la composición de los encuestados:

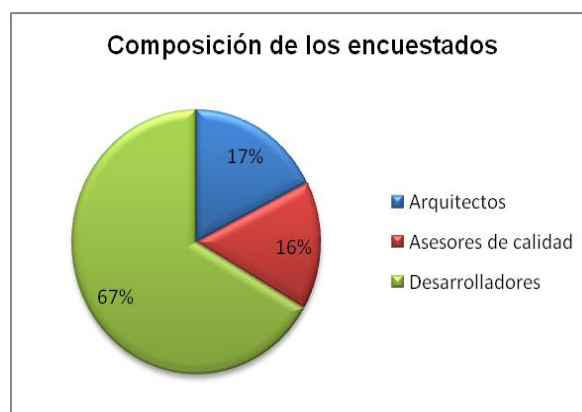


Fig. 6 Composición del personal encuestado (Ver Anexo 2)

Para el diseño de la encuesta fue tomada como variable global el proceso de evaluación de arquitectura, la cual se dividió en tres indicadores, factores de calidad, herramientas de evaluación y documentación del proceso. Cada unos de estos indicadores fueron a su vez, divididos en subindicadores, de los cuales se expresan los resultados a continuación:

Realización del proceso de evaluación de arquitectura en los proyectos productivos.(Sí se evalúa / No se evalúa)



Fig. 7 Resultados de la encuesta en la realización del proceso de evaluación (Ver Anexo 2)

Momento en que se evalúa la arquitectura. (Antes de establecido el diseño arquitectónico / Después / Ambos momentos)

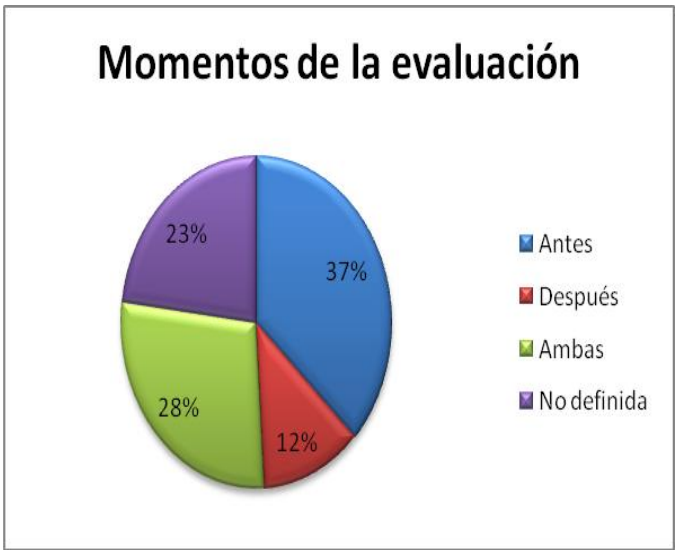


Fig. 8 Resultados de la encuesta en los momentos en que se evalúa la arquitectura (Ver Anexo 2)

Factores de calidad que se tuvieron en cuenta para realizar la evaluación arquitectónica. (Modificabilidad / Eficiencia / Confiabilidad / Funcionalidad / Usabilidad / Portabilidad / Ninguno)

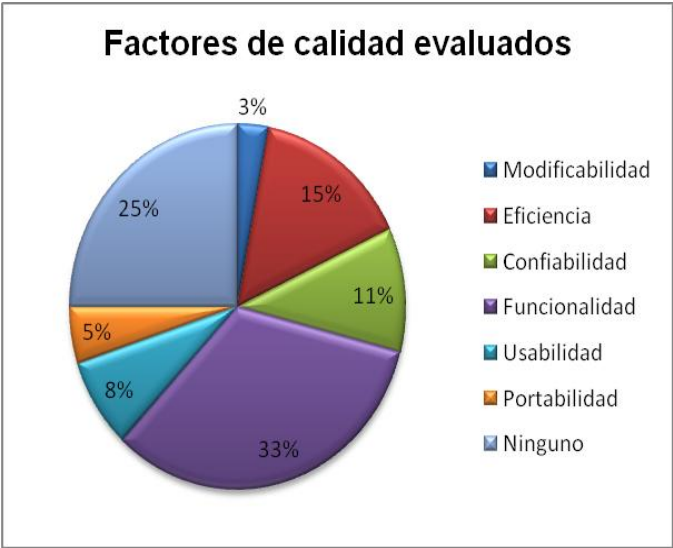


Fig. 9 Resultados de la encuesta con los factores de calidad que se evalúan (Ver Anexo 2)

Uso de modelos de calidad del producto para la selección y refinamiento de los factores de calidad evaluados.(Si / No uso modelo).

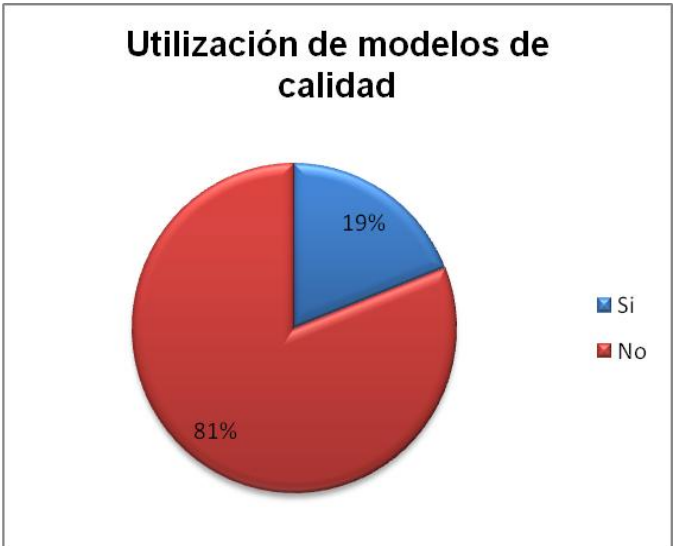


Fig. 10 Resultados de la encuesta en la utilización de modelos de calidad (Ver Anexo 2)

Uso de métodos de evaluación de arquitectura. (Si / No uso métodos)

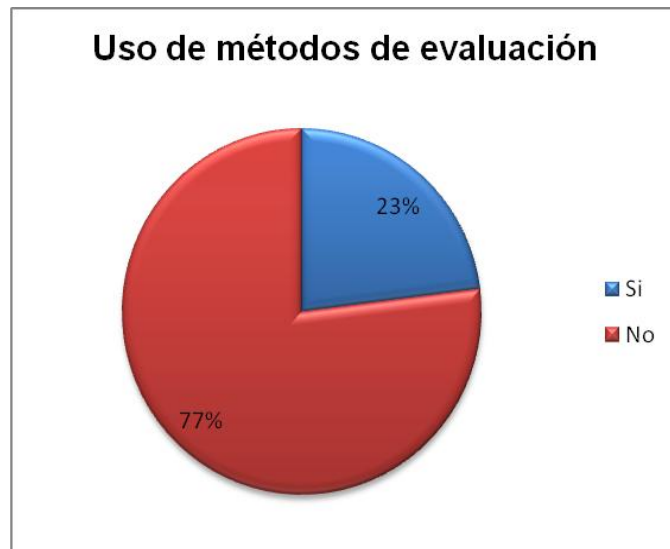


Fig. 11 Resultados de la encuesta en el uso de métodos de evaluación (Ver Anexo 2)

Métodos de usados en el proceso de evaluación arquitectónica. (Ninguno / SAAM / ATAM / CBAM / ALMA / FAAM / Otros)

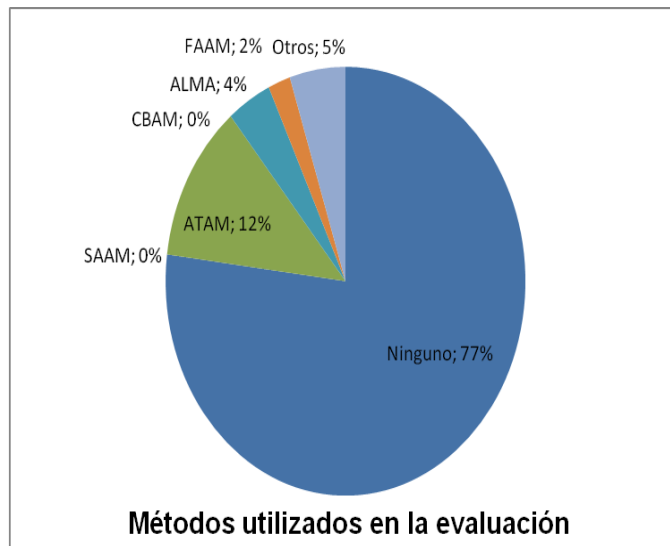


Fig. 12 Resultados de la encuesta en los métodos de evaluación utilizados (Ver Anexo 2)

Documentación de la evaluación de la arquitectura. (Se documenta / No se documenta)

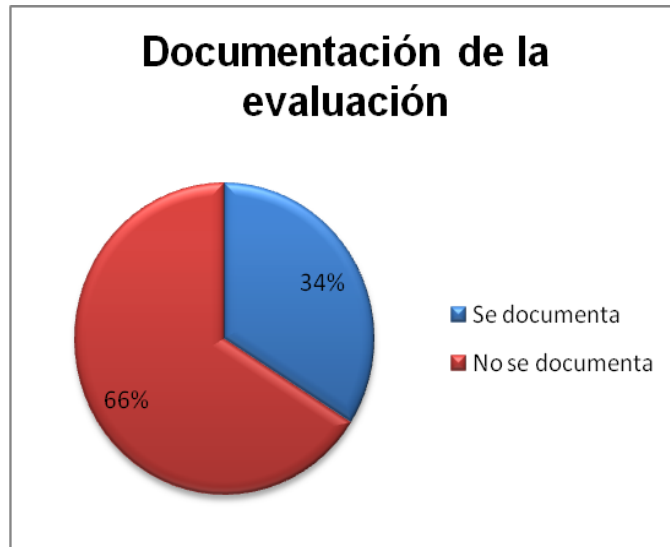


Fig. 13 Resultados de la encuesta en la realización del proceso de documentación (Ver Anexo 2)

Uso de plantillas que establezcan los elementos a documentar en el proceso evaluativo. (Si uso / No uso)



Fig. 14 Resultados de la encuesta en el uso de plantillas en la documentación (Ver Anexo 2)

Las encuestas realizadas arrojaron en primer lugar que solo el 37% de los proyectos que conforman la muestra realizan un proceso de evaluación de la arquitectura de software. Sin embargo, cabe destacar que de este 37%, un 23% no tienen definida la etapa de la evaluación, un 81% no utiliza un modelo de calidad para la selección

de los factores de calidad a medir, un 77 % no utiliza métodos de evaluación y un 66 % no documenta el proceso.

Esto demuestra que a pesar de que muchos aseguran evaluar la arquitectura de sus proyectos, no cuentan con herramientas para realizarla, y en el peor de los casos tienen desconocimiento de qué elementos abarca una evaluación arquitectónica.

Es observable, de que a pesar de la poca utilización de modelos de calidad para la selección de los factores de calidad, se muestran resultados de que muchos tienen en cuenta a la hora de evaluar la arquitectura, asociando fundamentalmente el factor de funcionalidad, relacionado con la aptitud del sistema de realizar las funciones para lo que fue desarrollado.

De manera general se identificaron y cuantificaron las siguientes dificultades:

- ✓ Un porcentaje considerable de los proyectos productivos de la universidad no realiza la evaluación arquitectónica.
- ✓ Existe desconocimiento de los elementos que abarcan un proceso de evaluación, factores de calidad que se evalúan, métodos de evaluación de arquitecturas y documentación del proceso para replicar experiencias.

Estas dificultades influyen directamente en el logro de la calidad del producto final, en la realización de un proceso de desarrollo eficiente y en la reutilización de componentes.

3.4 Análisis de los resultados de evaluación técnica del modelo.

Para organizar la evaluación por el comité de expertos se calculó la cantidad de evaluadores necesarios utilizando la expresión:

$$n = \frac{p * \left(\frac{1}{1-p} \right) * k}{i^2}$$

Para los valores de $i=0.10$, $p=0.01$, $K=6.6564$.

El resultado fue que deben participar en la evaluación un total de 7 expertos.

Posteriormente los expertos emitieron su juicio para darles peso a los indicadores. Se elaboró la tabla de los valores del peso relativo de cada criterio (Ver Anexo 4.2), la tabla para el cálculo de concordancia entre estos expertos (Ver Anexo 4.3).

Luego de haber calculado la dispersión ($S = \sum (\sum E - \sum \sum E / C)^2$), el coeficiente de concordancia de Kendall ($W = S / E^2 (C^3 - C) / 12$) y el X_2 real ($X_2 = E (C-1) W$), se realizó una comparación con los resultados de las tablas de estadígrafo X_2 .

$$X_{2\text{real}} < X_2(\alpha, c-1)$$

$$\text{Si } 1 - \alpha = 0.99$$

Donde α es el error

$$\alpha = 0,01$$

$$17,7888192 < 27,688$$

Como X_2 real es menor que el valor correspondiente en las tablas estadísticas, se puede asegurar que existe concordancia en el juicio emitido por los expertos. Por último se construyó la tabla de calificación de cada criterio (Anexo 4.4).

Luego de calculado el Índice de Aceptación dio como resultado 0,71357143, por lo que existe gran posibilidad de éxito de la propuesta.

La mayoría de los expertos que participaron en la valoración técnica de la propuesta plantean la necesidad e importancia del proceso de evaluación de arquitectura, la actualidad de la investigación y el aporte significativo que realiza al proceso de desarrollo de los productos de software de la universidad.

Algunos encuestados recomiendan la publicación de la investigación con vistas a que toda la comunidad de desarrollo de la universidad aplique la metodología en cada uno de sus proyectos. Esta acción contribuiría a probar de manera práctica la metodología propuesta, lo que permitiría corregir las posibles insuficiencias.

Plantean que una evaluación arquitectónica exitosa mejoraría la claridad de todo el equipo de desarrollo sobre la interacción entre componentes del producto, aumentaría el nivel de reutilización, ayudando a aumentar la eficiencia del equipo, a disminuir tiempo, costo y esfuerzo en el desarrollo, y a aumentar la calidad del proceso y del producto final.

Los encuestados recomiendan profundizar en los siguientes temas relacionados con el proceso de evaluación de la arquitectura de software:

- Cómo potenciar la utilización de estilos y patrones arquitectónicos en el diseño de las arquitecturas de los proyectos productivos de la UCI.

- Cómo utilizar los lenguajes de descripción arquitectónicos en los diseños de las arquitecturas de software.
- Proponer un conjunto de métricas e indicadores que permitan evaluar todos los atributos de calidad relacionados con la arquitectura de software.

3.5 Análisis de los resultados de la utilización de la metodología en un proyecto piloto.

En la línea de producción de contenidos audiovisuales se desarrollan en la universidad un conjunto de productos para diferentes entidades en el marco nacional y en el exterior del país. Luego de las experiencias y resultados del proyecto UCITV, quien elaboraba todos los materiales y la producción del canal interno de la universidad, se decide crear el polo de Video y Sonido digital, perteneciente a la facultad # 9.

Este polo enmarca un conjunto de proyectos que trabajan en la confección de materiales audiovisuales, catalogación, diseños y realización de medias, entre otras áreas temáticas. Uno de los proyectos más importantes dentro del polo es el de Captura y Catalogación de Medias, el cual tiene como alcance la filmación, diseño y catalogación de documentales de apoyo a una actividad específica, como, las actividades académicas, las actividades políticas y sociales de la universidad, las actividades deportivas, culturales, recreativas, entre otras.

Con el objetivo de probar la eficacia de la metodología, se seleccionó, por la etapa del ciclo de desarrollo en que se encontraba, las características propias de su arquitectura, la participación de todo el equipo del proyecto en el desarrollo de la presente investigación; el proyecto de Captura y Catalogación de Medias como piloto.

Otros de los factores tenidos en cuenta para la selección fue un conjunto de deficiencias identificadas mediante las entrevistas realizadas al personal del proyecto, entre ellas mencionar:

- ✓ Existieron incongruencias a la hora de la definición de la arquitectura del sistema, en especial al valorar varias alternativas.
- ✓ No se identificaba la necesidad de que los asesores de calidad participaran en la valoración de las arquitecturas candidatas.

- ✓ No se tomó en cuenta en el establecimiento de un diseño arquitectónico la valoración de los factores de calidad del producto.
- ✓ No se realizó ninguna documentación de las alternativas arquitectónicas valoradas ni se justificó la selección.
- ✓ Aparte de los dos arquitectos que trabajaron en las propuestas, el resto del equipo del proyecto desconocía la existencia de una arquitectura del sistema.
- ✓ No hubo participación de involucrados externos en la decisión arquitectónica.

En el momento en que se realizó la evaluación, el proyecto se encontraba en la fase de Elaboración, según la metodología de desarrollo RUP (Rational Unified Process) donde ya se había definido un diseño arquitectónico. El proceso de evaluación se realizó con los siguientes objetivos:

- ✓ Verificar si la arquitectura seleccionada cumplía con los factores de calidad establecidos por la norma ISO/IEC 9126-1 (ISO/IEC-9126-1, 2001).
- ✓ Identificar todos los puntos críticos de la arquitectura del sistema.
- ✓ Identificar posibles mejoras arquitectónicas.
- ✓ Vincular a gran parte del equipo del proyecto a la definición y evaluación de la arquitectura del sistema.
- ✓ Lograr que el personal del proyecto identifique las ventajas del diseño y la evaluación arquitectónica.
- ✓ Mejorar de manera general, el proceso de desarrollo del software y asegurar de esta manera la calidad del producto final.

El proceso de evaluación tuvo una duración de siete días, participando 12 personas. Se siguieron todas las recomendaciones realizadas en este trabajo, fundamentalmente una breve capacitación en los temas de refinamientos de los factores de calidad y diseño de una arquitectura de software.

Se hizo la presentación formal de la metodología, se discutieron cada un de las plantillas que conforman el expediente de evaluación, se presentó la adecuación del modelo ISO 9126 para efectos de la arquitectura de software

como producto intermedio del ciclo de desarrollo, se analizó un caso de estudio del mismo. Todas estas acciones posibilitaron subir el nivel técnico del equipo de evaluación y aumentar considerablemente la calidad del proceso.

Ya durante el proceso propio de la evaluación se tuvieron en cuenta los factores de calidad, funcionalidad, eficiencia, mantenibilidad, portabilidad y confiabilidad. Durante el proceso de refinamiento se utilizó el modelo ISO 9129 adecuado para la arquitectura de software, tomando un gran número de métricas e indicadores que propone el modelo para la valoración de los atributos de calidad. A continuación se presenta la arquitectura valorada:

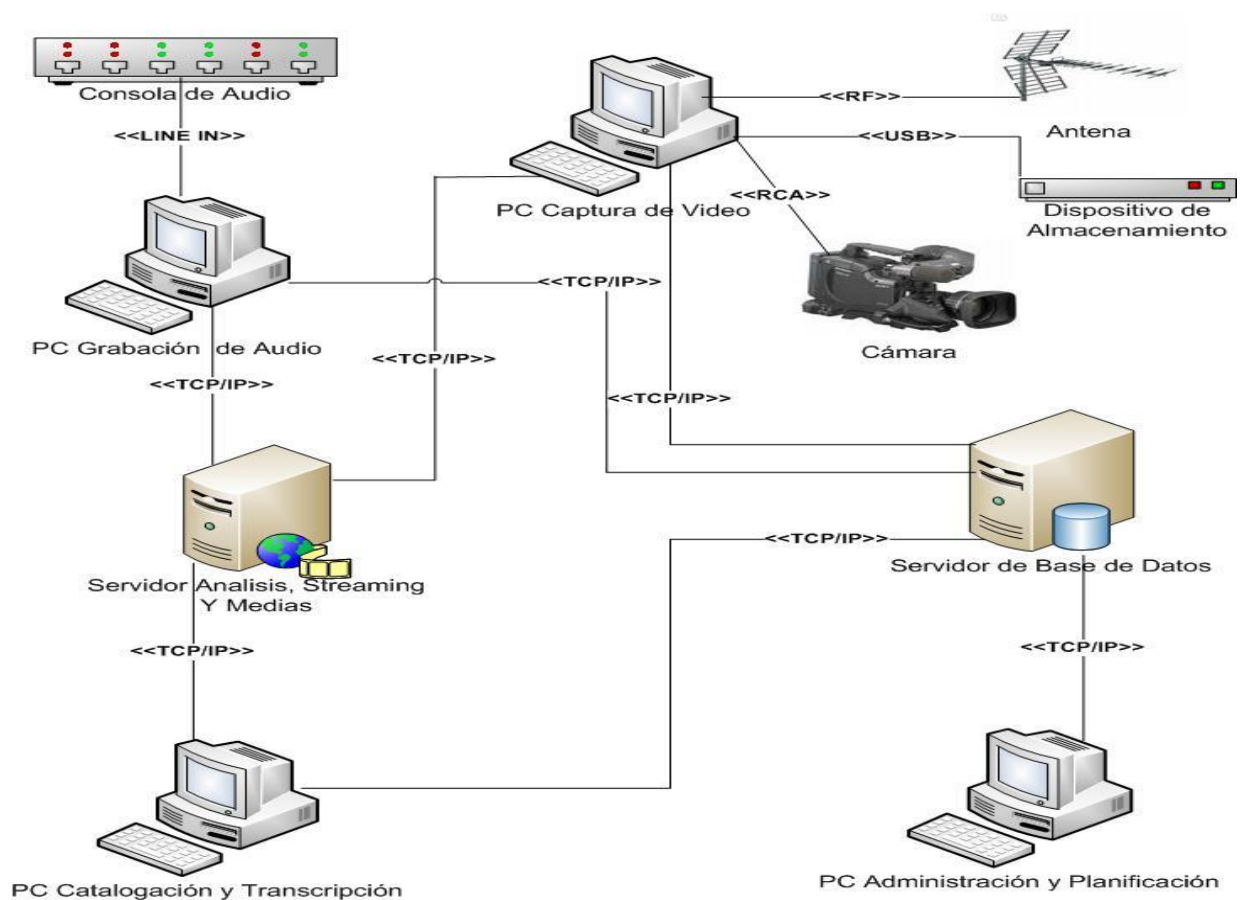


Fig 15 Arquitectura del Sistema de captura y catalogación de medias

Para la evaluación arquitectónica se definieron por parte de los implicados en el proceso, 17 escenarios. De estos se comprobaron aproximadamente 12 escenarios para el cumplimiento de la mayoría de los atributos de calidad definidos. Para más información consultar el Anexo 6.

En este proceso de evaluación se detectaron varios puntos de sensibilidad de la arquitectura, y se propusieron cambios arquitectónicos, entre los más importantes, el

cambio del Servidor de Streaming por su incompatibilidad con formatos de videos catalogados.

Se llevo a cabo un proceso eficiente de documentación, actualizando cada unas de las plantillas ante nuevos enfoques y decisiones. Se elaboró el documento final de las memorias del proceso evaluativo y se circuló a todos los integrantes del proyecto vía correo.

Para garantizar el nivel de partida y poder medir el impacto de la utilización de la metodología propuesta, se diseñó una encuesta (Ver anexo 3), la cual fue aplicada antes y después de la evaluación arquitectónica, a todos los que participaron. A continuación se brinda una comparación de los resultados antes y después de la aplicación de la metodología.

Importancia de la evaluación de la arquitectura de software en el proceso de desarrollo. (Baja / Media / Alta)

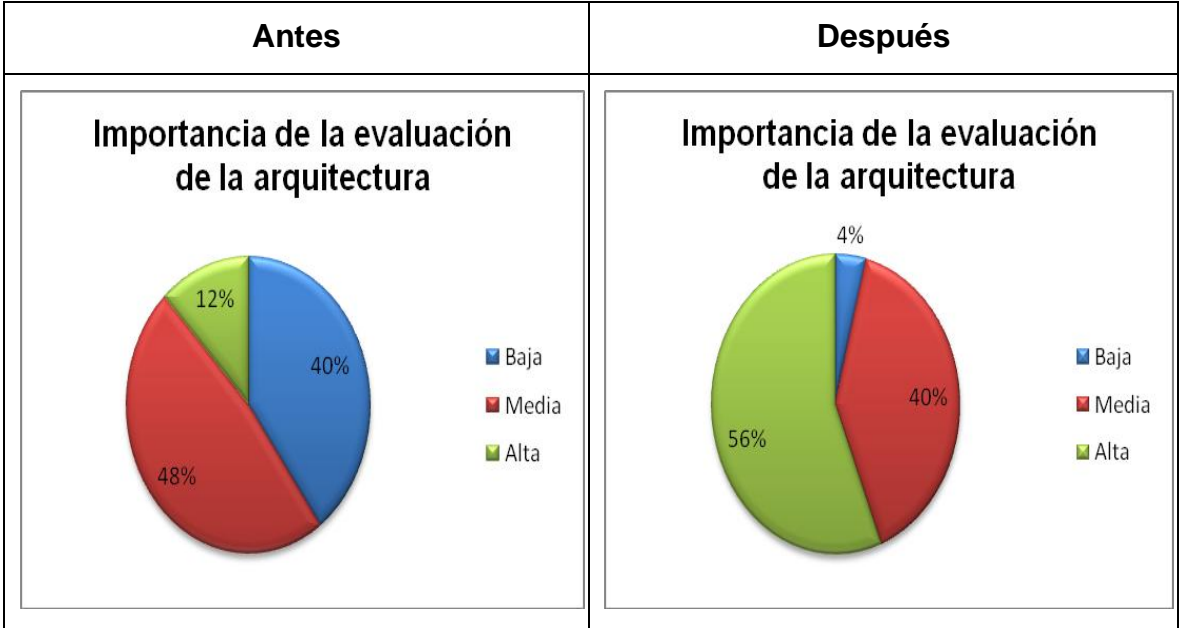


Fig. 16 Comparación de los resultados en el nivel de importancia de la evaluación arquitectónica (Ver Anexo 3)

Participantes en la evaluación de la arquitectura de software (Arquitectos / Líder de proyecto / Desarrolladores / Asesores de calidad / Clientes)

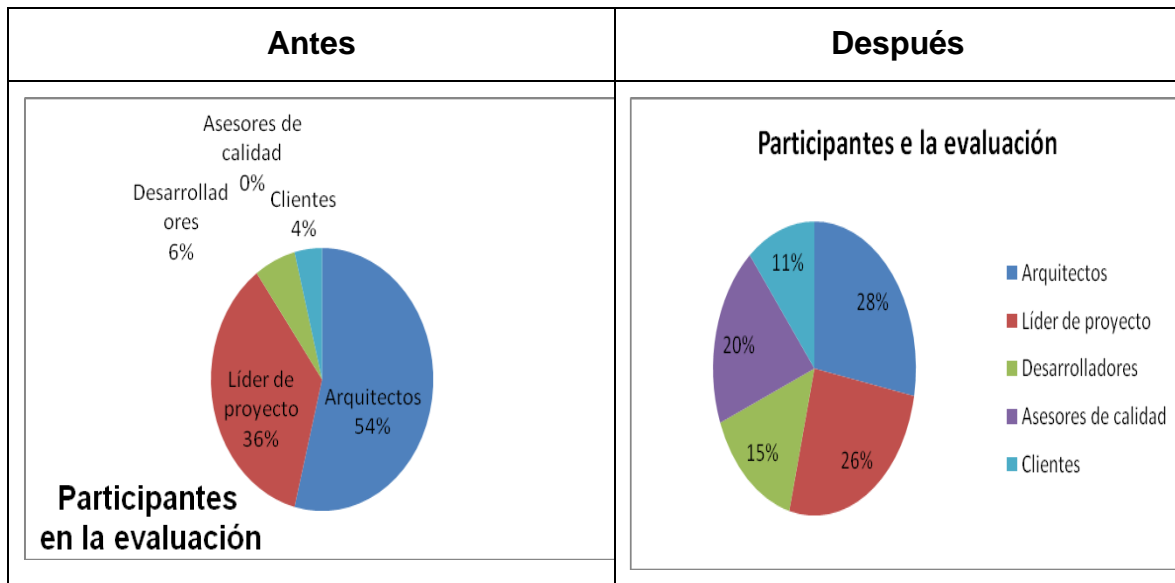


Fig. 17 Comparación de los resultados en la conformación de los equipos de evaluación (Ver Anexo 3)

Importancia de la documentación del proceso evaluativo (Baja / Media / Alta)

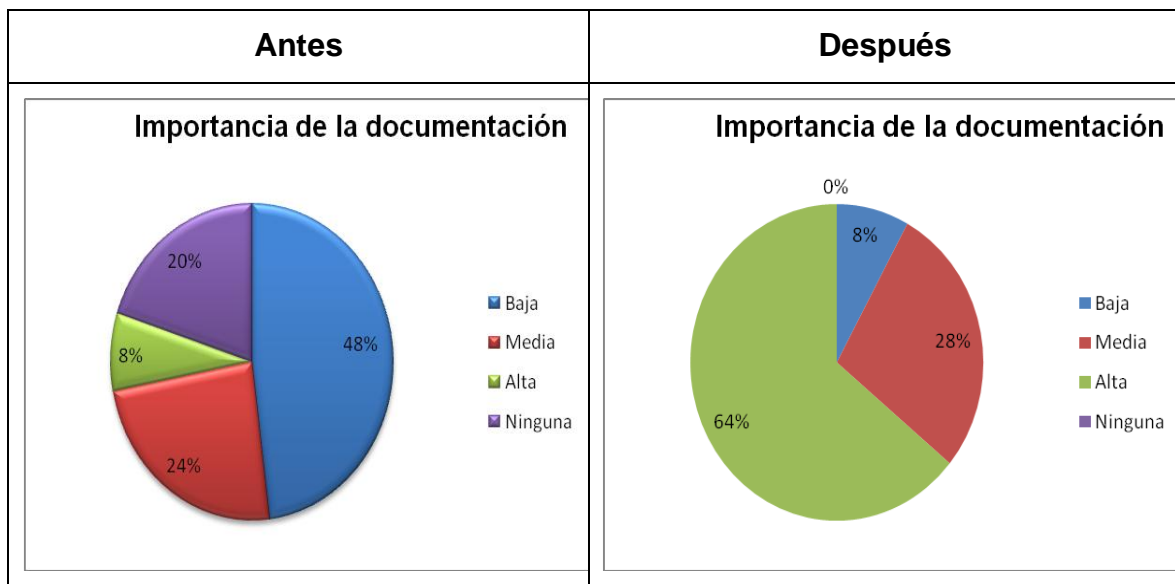


Fig. 18 Comparación de los resultados en el nivel de importancia del proceso de evaluación (Ver Anexo 3)

Importancia de la verificación de factores de calidad en la arquitectura del sistema durante el proceso de evaluación. (Baja / Media / Alta)

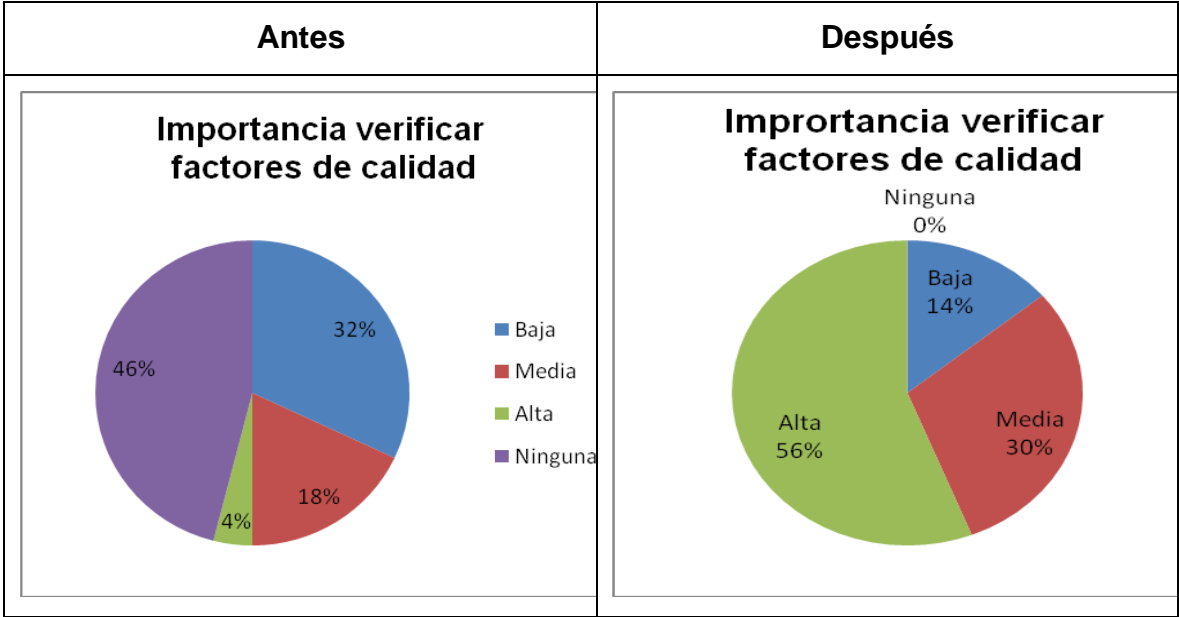


Fig. 19 Comparación de los resultados en el nivel de importancia de la verificación de los factores de calidad (Ver Anexo 3)

Como se puede apreciar luego de ser aplicada la metodología de evaluación de arquitectura de software, con todas las recomendaciones que propone esta tesis que se lleven a cabo durante el proceso de utilización, se muestran resultados. Estos no se evidencia solo desde el enfoque de las mejoras propuesta en la arquitectura diseñada, y la verificación de los atributos de calidad, sino que además el equipo evaluador ganó en conocimientos de que aspectos son de suma importancia en el desarrollo del proceso evaluativo, las ventajas que tiene la realización del mismo, cómo conformar los equipos de evaluación y la relevancia del proceso de documentación para la utilización de experiencias en proyectos similares.

Esto datos y el entendimiento logrado por parte de todo el personal del proyecto acerca de la arquitectura del sistema en desarrollo, logró agilizar el cronograma de trabajo en un total de 15 días. Luego de un tiempo de haber realizado la evaluación, el equipo de arquitectura realizó un listado de componentes y funciones comunes entre algunos módulos y estableció un repositorio de componentes en vista a evitar esfuerzos múltiples en el desarrollo y diseño de los mismos, y fomentar la reutilización.

3.6 Conclusiones del capítulo.

Durante este capítulo se proponen un conjunto de acciones con el objetivo general de garantizar la calidad del proceso de aplicación de la metodología de evaluación arquitectónica propuesta. Se analizan el resultado de la entrevista realizada y de las dos encuestas, reflejando los indicadores medidos y los resultados, mediante gráficos de pastel y porcentos. Se exponen los resultados arrojados por la aplicación del método de evaluación técnica de la propuesta, reflejando los cálculos de la dispersión, el coeficiente de Kendall, el Chi cuadrado real (X_2) y el índice de aceptación final de la propuesta. Finalmente se expone el proceso de aplicación de la metodología de evaluación propuesta en un proyecto piloto y los resultados mostrados luego de su aplicación.

Conclusiones y recomendaciones.

Conclusiones.

La metodología de evaluación de arquitectura de software:

- ✓ Contribuyó a elevar la calidad del proceso de desarrollo luego de establecido el diseño arquitectónico, y a su vez la calidad del producto.
- ✓ Promovió el intercambio entre clientes, equipo de arquitectura y desarrolladores, aumentando la eficiencia del proceso de desarrollo.
- ✓ Fomentó la reutilización de componentes, clases y funciones en los proyectos productivos.
- ✓ Aumentó los conocimientos técnicos sobre la arquitectura de software en el personal de los proyectos productivos.
- ✓ Permitió incluir la evaluación arquitectónica en el proceso de aseguramiento de la calidad de los proyectos de desarrollo de software.

Se realizó la evaluación técnica del modelo a través de un comité de expertos que auguró una probabilidad alta de éxito. Se diseñó un proceso de aplicación de la metodología de evaluación de arquitectura y se ejecutó su utilización en la evaluación de la arquitectura propuesta en un proyecto piloto, brindando resultados favorables.

Por tanto se concluye que el objetivo se cumple y se valida la hipótesis al proponer una metodología de evaluación de arquitectura de software en los proyectos productivos de la UCI contribuye a mejorar la calidad del proceso de desarrollo y del producto final.

Recomendaciones.

Los objetivos del trabajo no abarcan todos los elementos a definir; los cuales son amplios y diversos. Por lo que se propone:

- ✓ Definir posibles métricas y/o indicadores que permitan medir todos los atributos de calidad que impactan directamente en la arquitectura de software.
- ✓ Realizar un estudio de factibilidad económica que demuestre la factibilidad o no de este modelo.
- ✓ Aplicar, evaluar y revisar la implantación del modelo en proyectos de diversas líneas de producción de la UCI para su posterior implantación masiva.
- ✓ Incluir la actividad de evaluación de arquitectura de software en los lineamientos de calidad establecidos en la universidad.
- ✓ Automatizar el proceso de documentación que permita una mejor gestión del conocimiento de las experiencias de cada evaluación.

Bibliografía

- Berander, P, y otros. 2005.** *Software quality attributes and trade-offs*. s.l. : Blekinge Institute of Technology, 2005.
- DoD-STD-2168. 1985.** *Draft Military Standard, Software Quality Evaluation*. s.l. : Department of Defense, 1985.
- Dromey, G. 1996.** *Cornering the Chimera*. s.l. : IEEE Software, 1996.
- Gillies, A. 1997.** *Software Quality: Theory and Management*. London : Thomson Computer Press, 1997. Second edition.
- Grady, R y & Caswell, D. 1987.** *Software Metrics: Establishing a company-Wide Program*. s.l. : Prentice Hall, 1987.
- IEEE-729. 1983.** *IEEE Standard Glossary of Software Engineering Terminology*. s.l. : IEEE Std, 1983.
- ISO/IEC-9126-1. 2001.** *Software engineering - Product quality - Part 1: Quality model*. s.l. : ISO, 2001.
- ISO-8402. 1994.** *Quality management and quality assurance- vocabulary*. Genova : International Organization for Standarization, 1994.
- McCall, J., Richards, P. y & Waters, G. 1977.** *Factors in Software Quality*. s.l. : Rome Air Development Center, RADC-TR-77-369, 1977.
- . **1977.** *Factors in Software Quality*. Roma : Rome Air Development Center, 1977. RADC-TR-77-369.
- Pressman, R. 2002.** *Ingeniería de software. Un enfoque práctico*. s.l. : McGrawHill, 2002. 5ta Edición.
- Pressman, R. S. 1998.** *Ingeniería del software. Un enfoque práctico*. s.l. : McGrawHill, 1998. 4ª Edición.
- Barbacci, M. 2005.** *Quality Attributes*. s.l. : Carnegie Mellon University, 2005.
- Bass, y otros. 2005.** *Recommended Best Industrial Practice for Software Architecture Evaluation*. s.l. : Software Engineering Institute, 2005.
- Bosch, J. 2000.** *Design & Use of Software Architectures*. s.l. : Adison-Wesley, 2000.

Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. 2004. *Arquitecturas de Software*. 2004.

Carriere, J., Kazman, R. y Woods, S. 2007. *Toward a Discipline of Scenario based Architectural Engineering*. s.l. : Software Engineering Institute, Carnegie Mellon University, 2007.

Céspedes Vega, Anisleydi, y otros. 2007. *Procedimiento para Evaluar Arquitecturas de Software*. C. de la Habana : s.n., 2007.

Clements, P. 2006. *A survey of Architecture Description Languages*. s.l. : Software Engineering Institute, Carnegie Mellon University., 2006.
<http://www.sei.cmu.edu/publications/articles/survey-adl.html>.

Clements, Paul, Bergey, John y Mason, Dave. 2005. *Using the SEI Architecture Tradeoff Analysis Method to Evaluate WIN-T: A Case Study*. Software Engineering Institute : Software Architecture Technology Initiative, 2005.

Eskenazi, Evgeny y Fyukov, Alexander. 2004. *Quantitative Prediction of Quality Attributes for Component-Based Software Architectures*. Universidad de Eindhoven, The Netherlands : s.n., 2004.

Garlan, David y Shaw, Mary. 1994. *An Introduction to Software Architecture*. s.l. : <http://www.sei.cmu.edu/publications/documents/94.reports/94.tr.021.htm>, 1994. CMU/SEI-94-TR-021.

Hernández León, Rolando A. 2007. *Curso básico de Gestión de Proyecto*. C. de la Habana : s.n., 2007.

Ionita, Mugurel T., Hammer, Dieter K. y Obbink, Henk. 2006. *Scenario-Based Software Architecture Evaluation Methods: An Overview*. Eindhoven, The Netherlands, : SARA, 2006.

Kazman, R. 2006. *Tool Support for Architecture Analysis and Design*. Department of Computer Science, University of Waterloo. : s.n., 2006.

Kazman, R., Bass, L. y Clements, P. 2003. *Software Architecture in Practice*, Addison-Wesley,. 2003.

Kazman, R., Clements, Paul y Klain, M. 2005. *Evaluating Software Architectures. Methods and case studies*. s.l. : Addison Wesley, 2005.

Lane, Thomas G. 1990. *Studying Software Architecture Through Design Spaces and Rules*. s.l. : Software Engineering Institute, 1990.

Larsson, M. 2004. *Predicting Quality Attributes in Component-based Software Systems.* s.l. : Department of Computer Science and Engineering, Malardalen University, 2004. thesis.

Losavio, F, y otros. 2003. *Quality Characteristics for Software Architecture.* s.l. : JOT, 2003.

Mettala, Erik y Graham, Marc H. 1992. *The Domain-Specific Software Architecture Programs.* 1992. CMU/SEI-92-SR-009.

SARA. *Software Architecture Review and Assessment.*
[http://www.rational.com/media/products/rup/sara_rep] s.l. : Report Version 1.0 available in electronic.

Shaw, Mary. 1989. *Larger Scale Systems Require Higher-Level Abstractions.* IEEE Computer Society : s.n., 1989.

Shaw, Mary y Garlan, David. 1994. *Characteristics of Higher-Level Languages for Software Architecture.* Pensilvania : Software Engineering Institute, 1994. CMU/SEI-94-TR-023.

Anexos.

Anexo 1. Diseño de la entrevista realizada.

1. ¿Considera que en los proyectos de la universidad se evalúa la arquitectura de software?
2. ¿Considera que es necesario la realización de esta actividad? ¿Por qué?
3. ¿De qué forma se evalúa la arquitectura de software en su proyecto?
4. ¿En que etapa cree usted que debería evaluarse la arquitectura? ¿Por qué?
5. ¿Sabe cuáles son las técnicas de evaluación de arquitectura?
6. ¿Qué métodos de evaluación de arquitectura conoce?
7. ¿Qué sería beneficioso documentar al evaluar una arquitectura? ¿Por qué?
8. ¿Cree que si se lleva a cabo la evaluación de arquitectura se logrará algunas de estas opciones? Marque con una cruz las que considere.

Mejorar las decisiones arquitectónicas.

Predecir el comportamiento de los atributos de calidad antes cambios arquitectónicos.

Tomar un conjunto de decisiones de diseño en etapas tempranas del desarrollo.

Reducir riesgos en la fase posteriores al diseño arquitectónico.

Anexo 2. Diseño de la encuesta # 1.

La siguiente encuesta tiene como fin reunir estadísticas sobre la actividad de evaluación de arquitecturas en los proyectos y polos productivos de la UCI. Le pedimos sinceridad a la hora de responder las preguntas, le aseguramos confidencialidad y anonimato a su respuesta, solamente debe mencionar el rol que desempeña dentro del proyecto al que usted pertenece.

Rol: _____

Responder las siguientes preguntas y marcar con una x en el caso que haga falta:

PREGUNTAS

1- ¿Se evalúa la arquitectura de software en su proyecto productivo?

Si ___ No___

a). En caso afirmativo:

- ¿En que etapa del ciclo de desarrollo se ha evaluado la arquitectura en su proyecto?
 - i. Antes de definir la Arquitectura___
 - ii. Después de definir la Arquitectura ___
 - iii. En ambos momentos___
- Diga que factores de calidad se han evaluado fundamentalmente:
 - i. Modificabilidad
 - ii. Eficiencia___
 - iii. Confiabilidad___
 - iv. Funcionalidad___
 - v. Usabilidad___
 - vi. Portabilidad___
 - vii. Ninguno___
- ¿Utiliza algún modelo de calidad del producto en la selección y refinamiento de los factores de calidad?

- i. Si utilizo un modelo de calidad__
 - ii. No utilizo modelo de calidad__
- ¿Utiliza algún método de evaluación de arquitectura en el proceso evaluativo?
 - i. Si utilizo métodos de evaluación__
 - ii. No utilizo métodos de evaluación__
- Diga que métodos de los mencionados utiliza en la evaluación de la arquitectura de software de su proyecto.
 - i. SAAM__
 - ii. ATAM__
 - iii. CBAM__
 - iv. ALMA__
 - v. FAAM__
 - vi. Otros__
- ¿Se realiza la documentación de la evaluación de arquitectura en su proyecto?
 - i. Si se realiza__
 - ii. No se realiza__
- ¿Se utilizan plantillas para documentar la actividad de evaluación de arquitectura en su proyecto?
 - i. Si se utilizan plantillas__
 - ii. No se utilizan plantillas__

Anexo 3. Diseño de la encuesta # 2

La siguiente encuesta tiene como fin reunir estadísticas que permitan valorar el impacto de la utilización de la metodología de evaluación de arquitectura propuesta. Le pedimos sinceridad a la hora de responder las preguntas, le aseguramos confidencialidad y anonimato a su respuesta. Responder las siguientes preguntas y marcar con una x en el caso que haga falta:

1. ¿Qué importancia le da a la evaluación de la arquitectura de software ciclo de desarrollo de un producto de software?
 - a) Baja__
 - b) Media__
 - c) Alta__

2. ¿Qué roles de un proyecto conformarían a su consideración un equipo de evaluación de arquitectura de software?
 - a) Arquitectos__
 - b) Líder de proyecto__
 - c) Desarrolladores__
 - d) Asesores de calidad__
 - e) Clientes__

3. ¿Qué importancia le brinda a la documentación de la evaluación de la arquitectura de software en un proyecto de desarrollo de software?
 - a) Baja__
 - b) Media__
 - c) Alta__

4. ¿Qué importancia tiene para usted la verificación de los factores de calidad en el proceso de evaluación de arquitectura de software?
 - a) Baja__
 - b) Media__
 - c) Alta__

Anexo 4. Modelos y tablas para la evaluación técnica.

Anexo 4.1. Modelo No. 1 Guía para informar el peso de los criterios.

Fecha de recepción: 20/11/2008

Fecha de entrega: 10/01/2009

Nombre y Apellidos del evaluador: _____

• Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar:

- Grupo No.1..... 20
- Grupo No.2..... 35
- Grupo no.3..... 20
- Grupo No.4.....25

Para que el peso total asignado sea 100.

Grupo No. 1: Criterios de mérito científico

1. Calidad de la investigación.

Peso___

2. Novedad científica.

Peso___

3. Valor científico de la propuesta.

Peso___

4. Aporte científico.

Peso___

Grupo No. 2: Criterios implantación

5. Satisfacción de las necesidades de la producción.

Peso___

6. Garantía de principios básicos de la arquitectura y la calidad de software.

Peso___

7. Uso de técnicas y métodos de evaluación de arquitectura.

Peso____

8. Necesidad del empleo de la metodología.

Peso____

Grupo No.3: Criterios de generalización

9. Atractividad para su uso.

Peso____

10. Adaptabilidad a diferentes entornos de desarrollo.

Peso____

Grupo No.4: Criterios de impacto

11. Repercusión en la calidad del software.

Peso____

12. Comunicación entre los involucrados

Peso____

13. Mitigación de riesgos en etapas tempranas del desarrollo.

Peso____

14. Ventajas competitivas.

Peso____

15. Posibilidades de aplicación.

Peso____

Anexo 4.2 Modelo No. 2 Guía para la evaluación de cada criterio.

Fecha de recepción: 20/11/2008

Fecha de entrega: 10/01/2009

Nombre y Apellidos del evaluador: _____

- Criterios de medida que se evalúan en una escala de 1 - 5

Grupo No. 1: Criterios de mérito científico

1. Calidad de la investigación.

Evaluación____

2. Novedad científica.

Evaluación____

3. Valor científico de la propuesta.

Evaluación____

4. Aporte científico.

Evaluación____

Grupo No. 2: Criterios implantación

5. Satisfacción de las necesidades de la producción.

Evaluación____

6. Garantía de principios básicos de la arquitectura y la calidad de software.

Evaluación____

7. Uso de técnicas y métodos de evaluación de arquitectura.

Evaluación____

8. Necesidad del empleo de la metodología.

Evaluación____

Grupo No.3: Criterios de generalización

9. Atractividad para su uso.

Evaluación____

10. Adaptabilidad a diferentes entornos de desarrollo.

Evaluación___

Grupo No.4: Criterios de impacto

11. Repercusión en la calidad del software.

Evaluación___

12. Comunicación entre los involucrados

Evaluación___

13. Mitigación de riesgos en etapas tempranas del desarrollo.

Evaluación___

14. Ventajas competitivas.

Evaluación___

15. Posibilidades de aplicación.

Evaluación___

- Categoría final del proyecto

___ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

___ Bueno: Novedad científica, resultados destacados.

___ Aceptable: Suficientemente bueno con reservas.

___ Cuestionable: No tiene relevancia científica y los resultados son malos.

___ Malo: No aplicable.

- Valoración final

- Sugerencias del evaluador para mejorar la calidad del proyecto

- Elementos críticos que deben mejorarse.

Anexo 4.3 Tabla de los valores del peso relativo de cada criterio

G	C / E	E₁	E₂	E₃	E₄	E₅	E₆	E₇	E_p
20	C₁								0
	C₂								0
	C₃								0
	C₄								0
		0	0	0	0	0	0	0	0
35	C₅								0
	C₆								0
	C₇								0
	C₈								0
		0	0	0	0	0	0	0	0
20	C₉								0
	C₁₀								0
		0	0	0	0	0	0	0	0
25	C₁₁								0
	C₁₂								0
	C₁₃								0
	C₁₄								0
		0	0	0	0	0	0	0	0
T		0	0	0	0	0	0	0	0

Anexo 4.4 Tabla para el cálculo de la concordancia

Expertos/Criterios	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	∑E	Ep	ΔC	ΔC ²
C ₁								0	0	0	0
C ₂								0	0	0	0
C ₃								0	0	0	0
C ₄								0	0	0	0
C ₅								0	0	0	0
C ₆								0	0	0	0
C ₇								0	0	0	0
C ₈								0	0	0	0
C ₉								0	0	0	0
C ₁₀								0	0	0	0
C ₁₁								0	0	0	0
C ₁₂								0	0	0	0
C ₁₃								0	0	0	0
C ₁₄								0	0	0	0
DC	0	0	0	0	0	0	0	0	0	0	0
M∑E	0										
W	0										
X ²	0										

Donde:

Ep: Puntuación promedio de cada criterio

∑E: Sumatoria de las puntuaciones de cada criterio

M∑E: Media de los ∑E

ΔC: Diferencia entre ∑E y M∑E

C: Número de criterios

Anexo 4.5 Tabla de calificación de cada criterio

Criterios	Calificación (c)					P	P x c
	1	2	3	4	5		
C₁						0	
C₂						0	
C₃						0	
C₄						0	
C₅						0	
C₆						0	
C₇						0	
C₈						0	
C₉						0	
C₁₀						0	
C₁₁						0	
C₁₂						0	
C₁₃						0	
C₁₄						0	
Total							0
IA	0						

Donde:

P: Peso de los criterios.

C: Criterio promedio concedido por los expertos.

Anexo 5. Plantillas propuestas para la evaluación de la arquitectura de software.

Plantilla #1: Definición de las Evaluaciones de Arquitectura de Software.

Definición de las Evaluaciones de Arquitectura de Software

<Rector>

<Nombre del Proyecto>
<Nombre del producto>
<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Identificar en qué fase del desarrollo es más propicio evaluar la arquitectura, teniendo en cuenta el objetivo específico de cada evaluación. Exponer los elementos que permitan discernir por parte de los participantes, la relevancia de la evaluación arquitectónica.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usarán.]

Descripción de Evaluaciones

[Abarca el número de evaluaciones a realizar, momento(s) en el ciclo de desarrollo en que se realizarán, una breve descripción del motivo de la evaluación y la justificación del momento seleccionado.]

Nombre de la Evaluación:	Ejemplo de Evaluación1
---------------------------------	------------------------

No. De la Evaluación:	12	Responsable:	Jefe de Arquitectura	
Participantes:	Líder, Equipo de Arquitectura, Clientes			
Momento del ciclo de Desarrollo del Sistema				
Flujos de Trabajo	Fases			
	Inicio	Elaboración	Construcción	Transición
Modelación del Negocio				
Requerimientos	x			
Análisis y Diseño				
Implementación				
Prueba				
Despliegue				
Configuración y Cambios				
Gestión de Proyecto				
Ambiente				
Breve descripción del motivo de la Evaluación:	Verificar la existencia de los atributos de calidad exigidos por los clientes, de manera adecuada.			
Justificación del momento seleccionado:	Es el momento donde se tiene la lista de requisitos funcionales y no funcionales, y es una primera aproximación a la ASW ha utilizar.			

Referencias

[\[Lista de evaluaciones definidas\]](#)

Número	Nombre de la Evaluación
[1]	Ejemplo 1
[..]	Ejemplo ...
12	Ejemplo de Evaluación1

Plantilla #2: Cronograma de Evaluaciones de Arquitectura.

Cronograma de Evaluación de Arquitectura

<Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Lograr organización y coherencia en las evaluaciones programadas. Además permite que los participantes organicen mejor su tiempo, y ocurran menores incidencias en la asistencia a las actividades evaluativas, que son de vital importancia.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usarán]

Cronograma de Evaluaciones

[Para el control del tiempo y del cumplimiento de las evaluaciones previstas; se registrará el número, el nombre, el lugar, las observaciones y las fechas de inicio y fin, de las evaluaciones.]

Nº	Nombre de la Evaluación	Fecha Inicio			Lugar	Tiempo		Fecha Fin			Observaciones
		D	M	A		Hor	Min	D	M	A	
		00	00	00		00	00	00	00	00	

2.1 Referencias

[Lista de documentos a los que se hace referencia]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Definición de los Involucrados en el Proceso de Evaluación <Interno>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Selección de los involucrados, externos e internos, que participarán en la evaluación. Así como la asignación a cada involucrado de su(s) misión(es) específica(s) durante la fase de presentación.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Roles y responsabilidades

[Enunciar los involucrados externos e internos, que participarán en la evaluación, registrando el rol que desempeñan, el nombre y apellidos así como las responsabilidades que tendrá en la evaluación, además se enumeraran para tener un control de cuantos involucrados existen]

Nº	Rol	Nombre y Apellidos	Responsabilidades en la Evaluación
1			
2			
3			
4			

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Plantilla #4: Definición del Grupo de Evaluación.

Definición del Grupo de Evaluación <Interno>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Selección del personal que guiará el proceso de evaluación y apoyará a los involucrados en las distintas definiciones que comprende la fase de desarrollo.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan]

Roles y Responsabilidades

[Enunciar el personal que guiará el proceso de evaluación y apoyará a los involucrados en sus misiones. Registrando el rol que desempeña, el nombre y apellidos, la cantidad de evaluaciones en las que ha participado anteriormente así como los resultados obtenidos en las experiencias de evaluaciones anteriores. Se registra además un número para el control de la cantidad de personas que guiarán la evaluación.]

Nº	Rol	Nombre y Apellidos	Cantidad de Evaluaciones Anteriores	Resultados Anteriores
1				
2				
3				
4				

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Plantilla #5: Definición de las Metas del Negocio.

Definición de las Metas del Negocio <Interno>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Diseñar todos los elementos que tributen información al análisis arquitectónico, la selección de atributos de calidad esperados y la definición de los escenarios, a exponer en la fase de presentación.]

Definiciones, Acrónimos y Abreviaturas.

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usarán]

Metas del Negocio

[Este artefacto abarca elementos como, arquitectura de información, descripción de los procesos que debe soportar el sistema con sus flujos alternativos, actores del negocio, etc. Todos estos elementos deben ser presentados con una panorámica arquitectónica, o sea de que forma impactan sobre le diseño arquitectónico y viceversa.]

Nombre del Proceso:	
Actores del Negocio:	
Descripción del Proceso del Sistema:	
Flujo Normal de Eventos del Proceso	
1.	2.
3.	4.

	4.1
Flujos Alternativos de los Procesos del Sistema	
	4.1
Arquitectura de la Información	
Formato de la Información Requerida para realizar el Proceso:	
Formato en que se deben mostrar los Resultados:	
Restricciones de la Información que se Maneja:	
Validaciones de la Información para que no colapse el Sistema:	
Medida del flujo de Información que circula por la red:	
Grado de disponibilidad que debe tener la Información:	

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Plantilla #6: Descripción de la Arquitectura.

Descripción de la Arquitectura <Interno>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Diseñar las presentaciones que tributen información al análisis arquitectónico, la selección de atributos de calidad esperados y la definición de los escenarios, a exponer en la fase de presentación.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan]

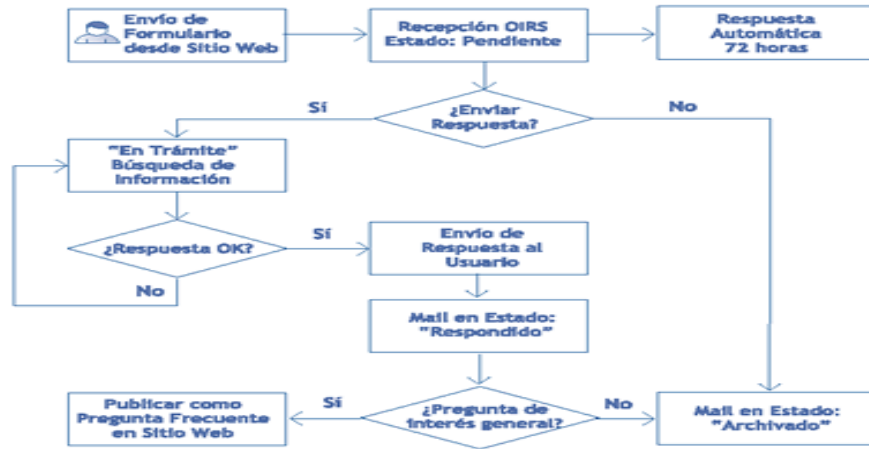
Descripción de los Componentes de la Arquitectura

[Descripción textual de cada uno de los elementos de la arquitectura, características, restricciones y riesgos en la arquitectura respecto a cambios en los componentes, así como la interacción entre componentes.]

Nombre de la Arquitectura:				
Principales Componentes				
Nombre Componente	Características	Restricciones	Riesgos Arquitectónicos	

Diagrama de Interacción entre Componentes

Ejemplo de diagrama de interacción:



Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Plantilla #7: Cronograma de Sesiones de la Evaluación.

Cronograma de Sesiones de la Evaluación

<Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Descripción de la forma en que se llevará a cabo el proceso de evaluación, número de sesiones, tiempo estimado de cada una, temas por sesiones, momento de exposición de los resultados, y responsables.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Cronograma de Sesiones.

[Para el control del tiempo y del cumplimiento de las evaluaciones previstas; se registrará en la siguiente tabla el número de sesiones estimadas, los participantes, el responsable, la fecha, el lugar y la hora de cada sesión, el tiempo estimado, el objetivo de cada sesión, los resultados esperados y las actividades que abarca cada sesión.]

Nº	Participantes	Fecha			Lugar	Hora		Responsable	Tiempo Estimado		Actividades que Abarca	Resultados Esperados
		D	M	A		Ho	Mi		Ho	Mi		

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Valoraciones de los Objetivos del Negocio <Interno>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Presentar toda la información preparada por los involucrados externos.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Valoraciones de los Objetivos del Negocio.

[Diseñar todos los elementos que tributan información al análisis arquitectónico, la selección de atributos de calidad esperados y la definición de los escenarios.]

Valoraciones de los Objetivos de Negocio	
Características del Negocio	Atributos de Calidad que Impactan en su Logro
Ejem: El 98% de los trabajadores que operarán el sistema, solo son operadores básicos de PC.	La facilidad de Uso, deberá ser máxima.

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Valoraciones de la Arquitectura Presentada

<Interno>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Realizar una presentación de la arquitectura establecida o el conjunto de arquitecturas candidatas, en dependencia del objetivo y momento de la evaluación.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Valoraciones de la Arquitectura Presentada.

[Registrar los elementos relevantes que aportó el debate, de todos los participantes en esta actividad, que sirva de referencia a actividades posteriores, en ella se registran las Valoraciones dadas, así como la especificación de si se logran o no los Atributos de Calidad (AC) que evalúan.]

Valoraciones de la Arquitectura Presentada		
Valoraciones	Logro de los AC que están evaluando.	
	Si	No

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Definición de los Atributos de Calidad <Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Seleccionar los atributos de calidad, que su comportamiento dependa directamente del diseño arquitectónico.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Definición de los Atributos de Calidad.

[Registrarán los atributos de calidad necesarios para el logro de la calidad requerida y que impacten directamente en el diseño arquitectónico. Para ello de cada atributo de calidad se recogerán datos como, tipo de atributo (sea dinámico o estático), una breve descripción, así como los elementos que deberá tener la arquitectura del software para satisfacerlo de la forma exigida por los clientes. Dentro de los atributos más exigidos por los clientes están Seguridad, Reutilización, Modificabilidad y Escalabilidad además de estos son evaluados también otros como: Confiabilidad, Desempeño, Facilidad de Uso, Robustez, Portabilidad, Disponibilidad, Mantenibilidad, Funcionalidad, Eficiencia.]

Atributo de Calidad:	Tipo de Atributo:
Breve Descripción del Atributo:	

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Definición de los Atributos de Calidad <Interno>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Describir el compartimiento que debe alcanzar cada uno de los atributos definidos, para lograr altos niveles de calidad en el software.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Definición de los Atributos de Calidad.

[Registrarán los atributos de calidad necesarios para el logro de la calidad requerida y que impacten directamente en el diseño arquitectónico. Para ello de cada atributo de calidad se recogerán datos como, tipo de atributo (sea dinámico o estático), una breve descripción, así como agrupación de atributos.]

Atributo de Calidad:			Tipo de Atributo:
Descripción de su Comportamiento			Elementos que debe tener la Arquitectura para logara satisfacer el Atributo:
Alto	Medio	Bajo	

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Grado de Prioridad de los Atributos de Calidad <Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Establecer prioridad de cada atributo definido.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Grado de Prioridad de los Atributos de Calidad.

[Abarca la organización de los atributos según su relevancia para el logro de la calidad. La prioridad de los atributos se establece en tres niveles, atributos de prioridad alta, atributos de prioridad media y atributos de prioridad baja. Los atributos de prioridad alta, son aquellos que se establecen directamente de las metas del negocio, y que los involucrados externos exigen con un alto rigor. Los atributos de prioridad media, son aquellos que establecen los involucrados internos, que ya van más orientados a cualidades técnicas que el sistema debe soportar, para en muchas ocasiones darle cumplimiento a los atributos de alta prioridad. Los atributos de baja prioridad, son aquellos que pueden ser establecidos por ambas partes, involucrados externos o internos, sin embargo, tienen un bajo impacto en el logro de la calidad exigida.]

Atributos de Calidad	Prioridad		
	Alta	Media	Baja

--	--	--	--

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Identificación de los Escenarios

<Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Identificar los escenarios que permita evaluar el comportamiento de los atributos definidos.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Identificación de los Escenarios.

[Abarca los escenarios que permiten evaluar el comportamiento de los atributos de calidad definidos. Haciendo una descripción detallada de cada escenario definido, teniendo en cuenta: nombre del escenario, origen del estímulo, estímulo, ambiente, componente, respuesta y medida de la respuesta Y documentando una breve descripción de los escenarios y atributo o atributos asociados al mismo.]

Nombre del Escenario:				
Origen del Estímulo:				
Estímulo	Ambiente	Componente	Respuesta	Medida de la Respuesta
Comportamiento de los Atributos de Calidad:				
Atributo:	¿Presente Cómo?		¿Ausente Por qué?	

Interacción entre los Actores del Sistema:		
Interacción entre Componentes Arquitectónicos:		

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Plantilla #14: Descripción detallada de los Escenarios.

Identificación de los Escenarios

<Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Realizar una descripción detallada de todos los elementos que conforman cada uno de los escenarios identificados.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Descripción Detallada de los Escenarios.

[Abarca los escenarios que permiten evaluar el comportamiento de los atributos de calidad definidos. Haciendo una descripción detallada de cada escenario definido, teniendo en cuenta: nombre del escenario, origen del estímulo, flujo de eventos, el comportamiento de los atributos de calidad vinculados al escenario.]

Nombre del Escenario:		
Origen del Estímulo:		
Flujo de Eventos:		
Comportamiento de los Atributos de Calidad:		
Atributo:	Se Cumple	No se Cumple

--	--	--

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Establecimiento de la Prioridad de los Escenarios

<Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito

[Establecer la prioridad de cada uno de los escenarios definidos.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Establecimiento de la Prioridad de los Escenarios.

[Abarca la organización de los escenarios según su relevancia para el logro de la calidad. La prioridad de los escenarios se debe establecer teniendo en cuenta los atributos que encierra y la prioridad de cada uno de ellos. En caso de que el escenario implique a más de un atributo, es importante observar la existencia de conflictos entre los atributos involucrados.]

Escenarios	Prioridad		
	Alta	Media	Baja

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Puntos Críticos de la Arquitectura

<Rector>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito.

[Determinar puntos de sensibilidad, puntos de riesgos y puntos de desventajas de la arquitectura diseñada.]

Definiciones, Acrónimos y Abreviaturas.

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Puntos Críticos de la Arquitectura.

[Determinar puntos de sensibilidad, puntos de riesgos y puntos de desventajas de la arquitectura diseñada.]

- Un punto de sensibilidad es la colección de componentes de la arquitectura que son fundamentales para un atributo de calidad en particular.
- Los puntos de desventajas es un punto de sensibilidad que es fundamental para el logro de múltiples atributos.
- Los puntos de riesgos son un subconjunto de los puntos de sensibilidad que puede evitar el logro de las metas de calidad del atributo.]

Puntos de Sensibilidad:			
N°	Punto	Breve Descripción	Atributos Involucrados

Puntos de Riesgo:			
N°	Punto	Breve Descripción	Atributos Involucrados
Puntos de Desventaja:			
N°	Punto	Breve Descripción	Atributos Involucrados

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Plantilla #17: Identificación de Posibles Cambios.

Identificación de Posibles Cambios
<Rector>

<Nombre del Proyecto>
<Nombre del producto>
<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito.

[Identificar posibles cambios o modificaciones que se puede aplicar a la arquitectura. Valorar el impacto de estos cambios sobre el comportamiento de los atributos de calidad.]

Definiciones, Acrónimos y Abreviaturas.

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Identificación de Posibles Cambios.

[Identificar posibles cambios o modificaciones que se puede aplicar a la arquitectura.

Valorar el impacto de estos cambios sobre el comportamiento de los atributos de calidad.]

N°	Propuesta de Cambio	Impacto en los Atributos de Calidad Involucrados			
		Atributo	Aumenta	Disminuye	No se Afecta

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Plantilla #18: Memorias del Proceso de Evaluación.

Memorias del Proceso de Evaluación <Entregable>

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha			Versión	Descripción	Autor
Día	Mes	Año			
00	00	0000	<0.0>	<detalles>	<nombre>

Propósito.

[Realizar las conclusiones del proceso evaluativo. Exponer los resultados obtenidos del proceso de evaluación de arquitectura de software.]

Definiciones, Acrónimos y Abreviaturas

[Identificar las principales definiciones, así como los acrónimos y abreviaturas que se usan.]

Memorias del Proceso de Evaluación.

[Este artefacto abarca un resumen de todos los aspectos relevantes del proceso, atributos, breve descripción de su comportamiento y prioridad, los escenarios y una breve descripción de cada uno].

Alto: Los escenarios con resultados positivos, dígame libre de errores. Deben estar en el rango de entre el 75 y el 100% de los establecidos para probar dicho atributo.

Medio: Los escenarios con resultados positivos, dígame libre de errores. Deben estar en el rango de entre el 74 y el 50% de los establecidos para probar dicho atributo.

Bajo: Los escenarios con resultados positivos, dígame libre de errores. Deben estar por debajo de él 50% de los establecidos para probar dicho atributo.

Atributo de Calidad:	Grado de Prioridad		
	Alta	Media	Baja

Tipo de Atributo:				
Breve Descripción del Atributo:				
Nombre del o los Escenarios Involucrados:				
Origen del Estimulo:				
Estimulo	Ambiente	Componente	Respuesta	Medida de Respuesta
Comportamiento del Atributo de Calidad				
Alto	Medio	Bajo		

Referencias

[Lista de documentos a los que se hace referencia.]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Documento 3

Anexo 6. Escenarios identificados durante le proceso de evaluación.

- El usuario solicita el **acceso al sistema**, en cualquier momento y el sistema se lo permite en **menos de 3 segundos**.
- El administrador solicita la **configuración de reportes**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El administrador solicita la **administración de reportes**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El administrador solicita la **configuración de procesos**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El administrador solicita la **administración de procesos**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El sistema **captura de un video**, en el momento planificado de forma automática en **1 segundo**.
- El operador de captura solicita la **captura de un video**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El Usuario de Catalogación solicita **editar una media**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El Usuario de Catalogación solicita **catalogar una media**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El cliente solicita la **catalogación de un video**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El Operador de Transcripción solicita **transcribir un audio**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El Operador de Captura solicita la **captura de un audio**, en cualquier momento y el sistema se lo permite en **menos de 5 segundos**.
- El cliente solicita el **cambio de servidor de base de datos**, en cualquier momento y el sistema se lo permite en **menos de 7 días**.

- El cliente solicita el **cambio de servidor de Streaming**, en cualquier momento y el sistema se lo permite en **menos de 7 días**.
- El cliente solicita el **cambio de la librería QT**, en cualquier momento y el sistema se lo permite en **menos de 7 días**.
- El cliente solicita el **cambio del Framework Hibernate**, en cualquier momento y el sistema se lo permite en **menos de 7 días**.
- El cliente solicita el **cambio de la consola de audio**, en cualquier momento y el sistema se lo permite en **menos de 7 días**.