

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/283459798>

MODELO PARA DISEÑAR ARQUITECTURAS DE SOFTWARE DURANTE EL DESARROLLO DE SISTEMAS DE INFORMACIÓN

Conference Paper · November 2011

DOI: 10.13140/RG.2.1.1121.9922

CITATIONS

0

READS

3,407

3 authors, including:



[Pedro Y. Piñero](#)

University of Information Sciences

101 PUBLICATIONS 177 CITATIONS

[SEE PROFILE](#)



[Nadia Porro Lugo](#)

University of Information Sciences

1 PUBLICATION 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Making decision in held problems [View project](#)



BusinessRedmine [View project](#)

MODELO PARA DISEÑAR ARQUITECTURAS DE SOFTWARE DURANTE EL DESARROLLO DE SISTEMAS DE INFORMACIÓN.

MSc. Michael González Jorrín^{1P}, Dr. Pedro Y. Piñero Pérez¹, Ing. Nadia Porro Lugo¹, MSc. René Lazo Ochoa¹

¹ Laboratorio de Gestión de Proyectos, Dirección Técnica de la Producción, Universidad de las Ciencias informáticas. Cuba. michael@uci.cu, nporro@uci.cu, ppp@uci.cu

Delegación de base: Universidad de las ciencias informáticas.

Temáticas del trabajo: Desarrollo de Aplicaciones, Modelos, Herramientas y Programación para la Web y Programación e Ingeniería de Software.

Resumen

La industria del software siempre necesita aumentar la productividad, disminuir costos y aumentar calidad. Contribuye a lograrlo la reutilización, si es debidamente explotada y soportada por la arquitectura del software, aprovechándose mejor así las oportunidades de negocio. El entendimiento de la arquitectura del software y del rol del arquitecto proporciona la sinergia necesaria dentro del equipo de desarrollo para lograr el éxito de los proyectos informáticos.

El presente trabajo presenta un modelo proactivo para el diseño de Arquitectura de Software aplicado a sistemas de información, que integra las corrientes más representativas, centrándose en el papel de este proceso en la gestión de los proyectos informáticos, la definición de los productos y su evolución. Aborda además la relevancia de los modelos arquitectónicos de referencia para estos sistemas a partir de su relación con la industria de software. En este modelo, soportado en vistas arquitectónicas, se especifica el Rol del Arquitecto de Software y se valoran los resultados obtenidos mediante análisis de los datos recopilados durante el proceso de aplicación modelo propuesto en tres casos de estudio de interés para la investigación.

Summary:

The software industry always needs to increase productivity, reduce costs and increase quality. Helps to achieve reuse, if properly operated and supported by the software architecture, and better advantage of business opportunities. Understanding the software architecture and the role of the architect provides the required synergy within the development team to ensure the success of IT projects.

This paper presents a proactive model for software architecture design applied to information systems, integrating the most representative trends, focusing on the role of this process in the management of IT projects, defining products and developments. Also addresses the relevance of the reference architectural models for these systems from its relationship with the software industry. In this model, supported by architectural views, specify the Role of Software Architect and evaluates the results obtained by analyzing data collected during the process of implementing the proposed model in three case studies of interest to the investigation.

Palabras claves: Arquitectura de software, Modelos arquitectónicos de referencia.

Keys words: Software architecture, Architectural models of reference.

Introducción

La industria del software no ha erradicado totalmente la producción artesanal. No siempre se sustenta la producción en modelos basados en la reutilización, que aumenten la productividad, disminuyan los costos y aumenten la calidad. Las arquitecturas de referencia como elemento fundamental para implantar ecosistemas productivos basados en la reutilización no son debidamente explotadas o entendidas, arriesgando la soberanía tecnológica. Muchas decisiones estratégicas sobre las tecnologías involucradas en los diseños arquitectónicos, impactan negativamente en el aprovechamiento de oportunidades de negocio.

Los SI constituyen un mercado, dada su complejidad y tamaño, para las medianas y grandes empresas desarrolladoras de software. Por su enfoque sistémico involucra como componentes a prácticamente a cualquier desarrollador más pequeño, forzando a la integración de sus soluciones y haciéndolas gravitar alrededor de las mismas arquitecturas de referencia.

La relevancia de los SI está en su capacidad para transformar las organizaciones actuales. Su uso facilita importantes mejoras en los procesos operativos, suministrando una plataforma de información necesaria para el monitoreo y control de las actividades, la toma de decisiones y, lo más importante, su implantación logra ventajas competitivas o reducir la ventaja de los rivales.

La perdurabilidad de estas aplicaciones está dada más por lo escalable de la solución, el rendimiento, la seguridad, la propia capacidad de ser mantenido y menos por cuantas funcionalidades tenga. Es interés del trabajo enmarcar una definición integral y sistémica de la Arquitectura de Software como: *la representación de alto nivel de abstracción de un sistema, está constituida por las partes del mismo, el mecanismo de integración e interrelación, restricciones y configuraciones a entornos de solución, los principios del diseño y evolución así como el ambiente tecnológico de implantación.*

Existen factores importantes a maximizar en la búsqueda del éxito de los proyectos: las competencias y la sinergia, en esta última tiene un gran impacto la comunicación entre los diferentes roles, esta es facilitada por la arquitectura pues constituye un lenguaje único entre los miembros del proyecto y como se conforma de decisiones técnicas, disminuyendo drásticamente los conflictos y la creatividad no controlada. (Ampuero, 2009)

Los responsables de estas decisiones técnicas y de sus consecuencias son los jefes de proyectos informáticos apoyados por los arquitectos de software, las competencias de estos últimos en los aspectos del desarrollo del producto deben ser más completas que el resto de los miembros del equipo (Ampuero, 2009) lo que le imprime un peso importante dentro de las decisiones.

Los profesionales de que se nutre la industria cubana provienen de un entorno donde la Arquitectura de Software, como disciplina, no se estudia en ninguno de los programas de carrera del país, se impone el auto aprendizaje para intentar cubrir, normar y dirigir todos sus aspectos (Ochoa, 2011). Esto provoca la improvisación durante la gestión de los proyectos.

Diagnóstico inicial sobre 6 productos

Utilizando la Universidad de las Ciencias Informáticas para este estudio se detectaron a través de entrevistas a líderes del producto A problemas: Bibliotecas gráficas mal aplicadas GTC y KT. Se rehizo todo el módulo, esta segunda versión se atrasó y provocó la pérdida de un contrato. Afectó los ingresos al tener que compartieron desarrollo con otro competidor que impuso otra tecnología. No se documentó debidamente la experiencia. Y hubo pérdidas de clientes potenciales por incompatibilidad del sistema operativo previsto para las estaciones de trabajo.

Para el producto B se observó como principal dificultad la falta de documentación referente a las experiencias tanto positivas como negativas. Además como el caso anterior se observó pérdida de clientes potenciales por incompatibilidad de sistema operativo para las estaciones de trabajo.

Otro el producto representado por F con el desarrollo de múltiples soluciones no coordinadas por diferentes estructuras de una misma organización. Además estas soluciones no han sido debidamente liberadas, poseen pobre documentación y divulgación convirtiendo estos esfuerzos en casos particulares y de pobre reutilización.

El patrón que describen los casos se puede observar en la tabla siguiente. Donde se analizaron:

- Fallas en el análisis: Roles, Responsabilidades, Capacitación, Metas, Metodologías y Procesos)
- Calidad: Competitividad, Satisfacción del cliente, Mejora continua
- Nivel de reutilización: como medida de la productividad en relación a además con la calidad.
- Costos: Impacto negativo en los costos, los ingresos o la independencia tecnológica.
- Tiempo: Retraso y afectaciones a compromisos.

Tabla 1. Datos resultantes del diagnóstico inicial en los casos de estudio.

A B C D E F	producto del diagnóstico	Fallas en el análisis								Calidad	Nivel de reutilización			Costos							Tiempo						
		Pruebas de concepto	Vista de Procesos	Vista de Presentación	Vista de Sistema	Vista de Bases de Datos	Vista de Integración	Vista de Tecnología	Vista de Infraestructura		Vista de Despliegue	Experiencia no documentada	Clases	Código	Documentación	Re hacer partes	Dificult. integración	Dificult. Despliegue	Hito de pago afectado	Costosas licencias para el cliente	Costosas licencias de desarrollo	Costos en capacitación y certificación	Desarrollo de componentes no previstos	Retrabajo	Duplicidad de esfuerzo	Retraso	Hito incumplido
	x	x		x	x						x	x	x	x	x												
	x			x							x	x	x	x													
					x	x			x	x				x	x	x					x				x	x	
	x	x	x		x	x	x	x			x	x	x	x	x		x	x	x	x	x		x		x	x	
	x	x			x	x	x	x	x												x	x	x		x		

En cuanto a los niveles de reutilización ningún caso llega a establecer un repositorio de activos, no reutilizan componentes, ni siquiera módulos se quedan a nivel de clases y código como se puede observar. Existen fallas generalizadas en las perspectivas del análisis arquitectónico por lo que se consideró iniciar la investigación.

El objetivo de este trabajo es presentar modelo proactivo de referencia basado en vistas como metodología para el diseño de Arquitectura de Software de sistemas de información, centrándose en el papel de este proceso en la gestión de los proyectos informáticos, la definición de los productos y su evolución.

Desarrollo

Las vistas arquitectónicas predominan en los diferentes marcos estudiados como factor común de elementos conceptuales conductivos en el diseño y desarrollo de la arquitectura de software (Bass, 2003), (Dargan, 1994), (Gamma, 1995.), (Buschmann, 1996). Se conceptualiza Vista Arquitectónica como *abstracción formalizada en prosa, lenguaje de modelado, gráfico o esquema informal, desde la que se describe las características arquitectónicas de un plano específico de la solución según el interés de involucrados*.

De los modelos de vistas arquitectónicas consultados se identifica la no existencia de un conjunto “canónico” de vistas. Existe diversidad de planteamientos de cómo formalizar las distintas perspectivas de un sistema (Kazman), (Bass, 2003), (Hofmeister, 2000), (Kruchten, 1995), (Reynoso, 2004) en ocasiones iguales pero identificadas con nombres diferentes y viceversa. Lo más adecuado sería seleccionarlas en función de satisfacer el interés de todos los involucrados.(Reed, 2001)

Los Modelos de desarrollo arquitectónico o *Architecture Development Method (ADM)* son otro punto de referencia asociados a elementos metodológicos, estrategias, modelos y las Guía de Referencia Arquitectónica. Existen organismos de estándares relacionados con la Arquitectura del software (ISO, CEN, IEEE, OMG) (Highsmith, 2001). Como resultado de estos trabajos se han obtenido especificaciones y recomendaciones de variada naturaleza, como RM-ODP, RUP, RDS, MDA, MOF, MEMO, XMI o IEEE 1471-2000 (Klein, 2003). También la mayoría de los framework y estrategias consultadas reconocen las vistas como la estructura central para la formalización y descripción de las arquitecturas.

Software Engineering Institute (SEI): provee técnicas y métodos para crear, documentar, evaluar y reconstruir una arquitectura robusta (SEI, 2011). Su estructura documental y metodológica y el enfoque a gestion de riesgo están muy completas.

En la literatura consultada se pueden distinguir cinco corrientes representativas, pero en ninguna de ellas se integran de manera sistémicas los elementos teóricos de las restantes. El presente trabajo se nutre e integra estas tendencias: Arquitectura como etapa de ingeniería y diseño orientado a objetos, Arquitectura estructural, basada en un modelo estático de estilos, ADLs y vistas, Arquitectura basada en patrones, Arquitectura procesual o de ciclo de vida y Arquitectura basada en escenarios.

Resultados

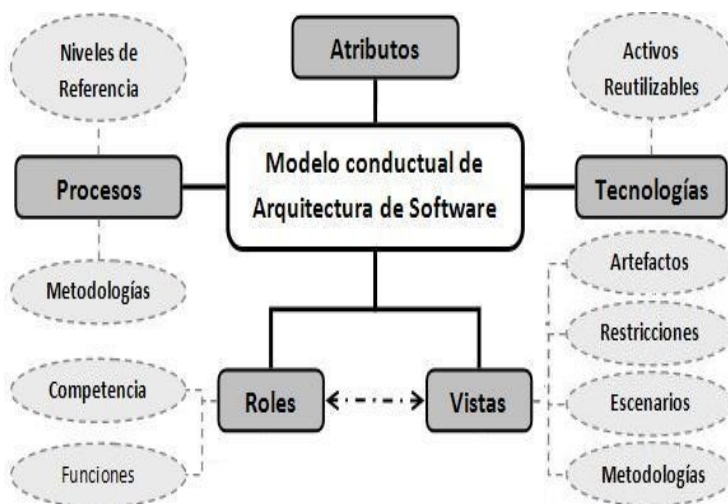


Figura 1 Modelo de referencia para el desarrollo arquitectónico basado en vistas.

El Modelo de referencia para el desarrollo arquitectónico basado en vistas (ver Figura 1) que se presenta se compone de dos tipos de elementos: los estructurales, que facilitan el entendimiento y análisis de la arquitectura y los de soporte a la gestión de proyectos que guían el desarrollo y permiten aprender haciendo.

En los elementos estructurales están las vistas, las tecnologías para el desarrollo y las reutilizables, y la observancia atributos de software.

Se formalizan estructuralmente las vistas arquitectónicas que conforman la solución de software. Las mismas están constituidas por los artefactos requeridos, las

restricciones tecnológicas o de diseño a valorar.

Entre los elementos de soporte están los procesos para el desarrollo de la arquitectura, los roles, metodologías, la guía base de análisis arquitectónico y una taxonomía preliminar de escenarios arquitectónicos tipos. El modelo incluye una formalización centrada en la especialización del rol del arquitecto de software, devenido de las diferentes áreas del conocimiento de la disciplina.

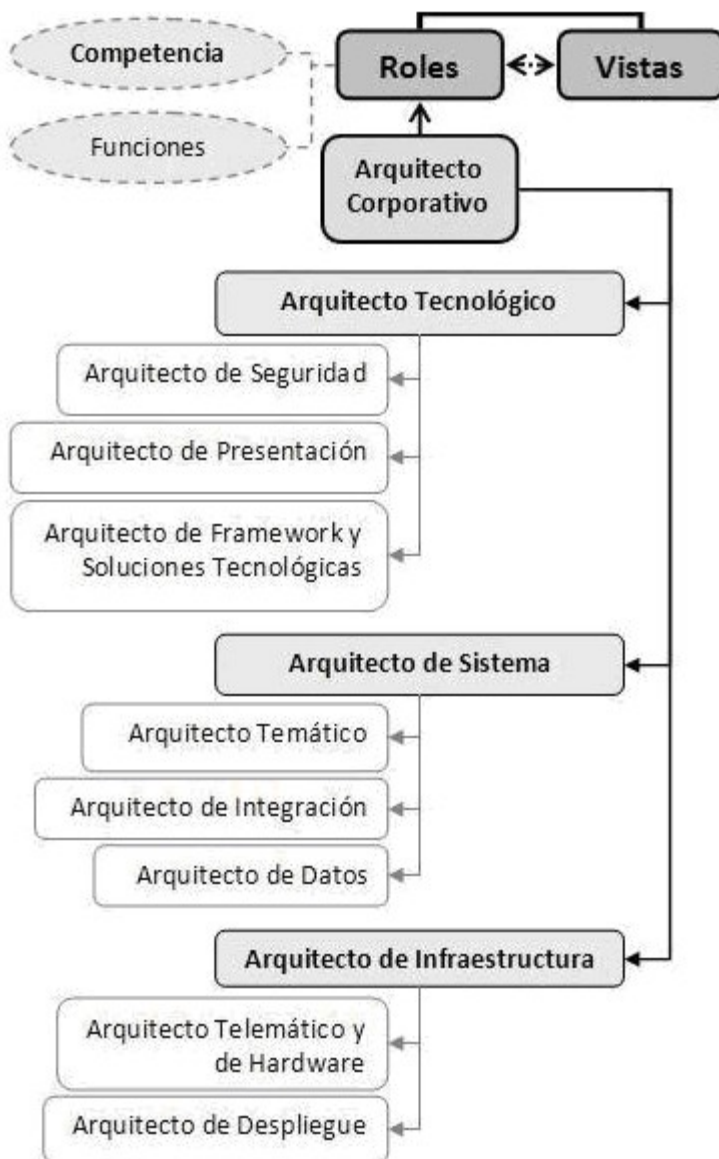


Figura 2 Roles del modelo de referencia para el desarrollo arquitectónico basado en vistas.

El rol del arquitecto en su especialización y definición es considerado entre los roles más complejos dentro de la ingeniería del software, responsable del desarrollo técnico del proyecto y por ende quien lo dirige desde esta dimensión. El arquitecto resulta el principal asesor del líder de software y orienta el camino estratégico de la organización de producción de software (Reynoso, 2004), (Escobar, 2005). El trabajo considera que la responsabilidad del arquitecto de software debe verse en cuatro dimensiones fundamentales una dimensión corporativa, responsable de predecir el alcance del producto, sus interioridades técnicas, las restricciones del diseño, la capacidad interoperabilidad e integración con otras plataformas, productos o el propio entorno. Y otras tres dimensiones representadas en el Arquitecto Tecnológico, el Arquitecto de Sistema y el Arquitecto de Infraestructura. (Ver figura 2)

La Guía base de análisis arquitectónico (Ver figura 3), desde la perspectiva de procesos, que facilita el ordenamiento y priorización de las vistas arquitectónicas, es además un elemento normativo, formativo y de auto evaluación del arquitecto. Conduce al arquitecto a través de las pruebas de concepto.

Facilidades de la guía base y las vistas.

La guía y el modelo en general impactan positivamente en la gestión del proyecto a través de los siguientes aspectos:

- Introducción de los conceptos básicos y metas técnicas del producto y por lo tanto determinación del alcance y la complejidad del proyecto.
- Definición y delimitación de las responsabilidades y competencias de los roles de la disciplina.
- o Facilidades para el trabajo colaborativo y en equipo, sobre objetivos comunes (atributos) y el mismo objeto (arquitectura).

- o Facilidades para la comunicación y el intercambio de información desde las diferentes perspectivas y niveles de abstracción.
- Establecimiento de reglas para el manejo de la guía base como metodología, potenciando el análisis del producto de diversas perspectivas minimizando los riesgos y asegurando la calidad del producto final así como una planificación más detallada.

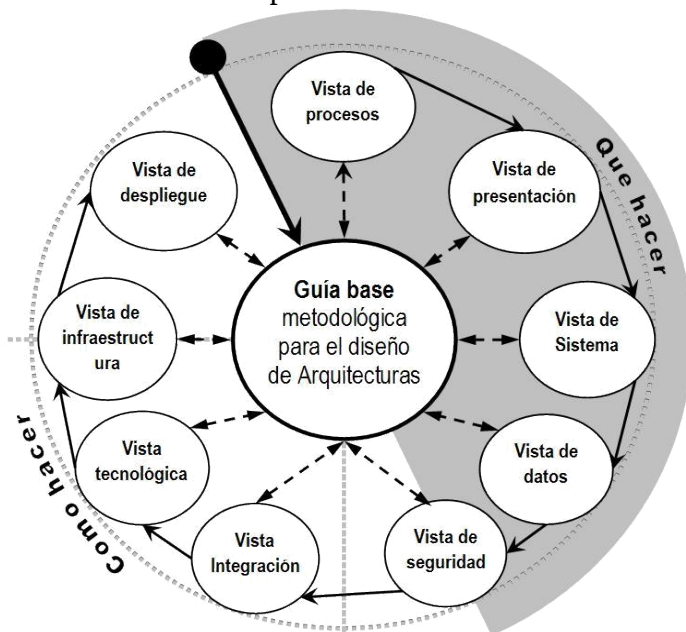


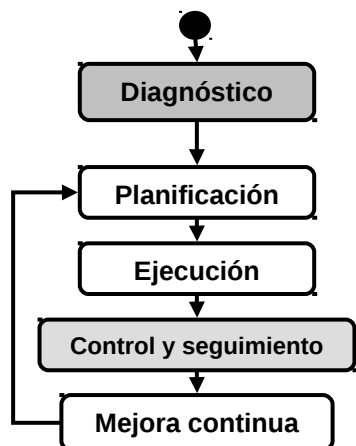
Figura 3 Guía base de análisis arquitectónico.

- Facilita el entendimiento del producto a desarrollar y acortar el tiempo de integración del equipo, pues garantiza un lenguaje común y la coordinación necesaria de las decisiones.
 - o Establecimiento de guías metodológicas durante el desarrollo.
 - o Análisis a través de diferentes perspectivas (vistas).
 - o Análisis, caracterización y clasificación de problemas.
 - o Suministro de patrones de solución para problemas tipo.
 - o Mecanismos de autoevaluación de metas por vistas, facilitando la retroalimentación.

- Incorporación, almacenamiento y socialización de información y conocimiento:
 - o Análisis de tendencias y regularidades para entornos colaborativos.
 - o Evolución de la experiencia del modelos expresada en soluciones y problemas tipos recolectados del entorno colaborativo.

Discusión y métodos

El modelo se introduce en tres entornos diferentes, los que constituyen nuestros casos de estudio, la introducción se hace a partir de una propuesta metodológica organizada en pasos (ver figura 4):



- Paso 1 Diagnóstico: Caracterización del entorno de aplicación, Determinación de la adherencia de las prácticas del entorno de aplicación respecto a la propuesta, Evaluación de los niveles de madurez de la infraestructura y el soporte.
- Paso 2 Planificación: Adaptación de la propuesta a las características del entorno, Desarrollo de instrumentos de captura de información sobre diseño arquitectónico, Establecimiento de un proyecto o plan de ejecución
- Paso 3 Ejecución: Capacitación y auto aprendizaje, Creación y maduración de las infraestructuras de soporte a la propuesta, Implantación de la propuesta y personalización de la propuesta

Figura 4 Metodología de implantación.

por dominios de aplicación y líneas de desarrollo, Establecimiento de mecanismos de apoyo y fiscalización.

- Paso 4 Control y seguimiento: Control y seguimiento de la generalización de la propuesta, Presentación de informes a la alta y media gerencia, Aplicación de auditorías a la arquitectura.
- Paso 5 Mejora continua: Registro de las lecciones aprendidas, Planificación de la mejora en la siguiente aplicación.

El Modelo de referencia para el desarrollo arquitectónico basado en vistas se aplicó en tres casos de estudio: El Caso 1 corresponde al “Proyecto ERP Cubano: CEDRUX” del centro CESGE de la Facultad 3 de la UCI (universidad de Ciencias informáticas), el Caso 2 corresponde al “Proyecto Viajero” a la Empresa SICS y el Caso 3 fue el proceso de generalización del modelo a los proyectos de toda la UCI. Los casos se aplicaron según orden de aparición perfeccionándose el modelo con cada aplicación.

Caso 1, un proyecto de gran envergadura en cuanto a recursos humanos y cantidad de requerimientos a implementar y contaba con un personal que tenía buena organización y una capacidad técnica media. Bajo estas condiciones, además de otras variables que también se midieron en el caso, se logró implantar las vistas propuestas por el modelo en un 83,33%. En una escala del 1 al 10: el impacto de la socialización del conocimiento fue de 8, el impacto de la productividad fue de 6, y el impacto de la estandarización del desarrollo fue 6.

Entre los resultados positivos que se arrojaron estuvo que se logró entrenar en las temáticas y conocimientos de la arquitectura de software a 42 profesionales y 67 estudiantes. La gestión de proyecto logró homogenizar las técnicas de dirección y toma de decisiones. Se logra integración del proyecto, simultaneidad las tareas, rotar los recursos tecnológicos y humanos en función de las prioridades. Alertó sobre de los principales riesgos (relacionados con RRHH y tecnologías). Facilitó el mantenimiento y la adaptación a las continuas modificaciones de requisitos.

Caso 2, un proyecto de mediana envergadura en cuanto a recursos humanos y cantidad de requerimientos a implementar, y contaba con un personal que tenía mala organización y una capacidad técnica mala. Bajo estas condiciones, además de otras variables que también se midieron en el caso, se logró implantar las vistas propuestas por el modelo en un 98,33%. En una escala del 1 al 10: el impacto de la socialización del conocimiento fue de 10, el impacto de la productividad fue de 10, y el impacto de la estandarización del desarrollo fue 10.

Entre los resultados positivos que se arrojaron estuvo que se logró entrenar en las temáticas y conocimientos de la arquitectura de software. Los recursos humanos (técnicos medio 47% y el resto recién graduados nivel superior) desconocían la existencia de esta disciplina del conocimiento. El proyecto Viajero con un atraso en cronograma de tres años y medio, mejoró su arquitectura en un 80%. El tiempo de estabilización del producto fue de 20 días una vez liberado la nueva versión. Los indicadores de respuesta de las consultas en tiempo real descendieron de 1.7 minutos a 3,57 segundos. La tecnología única y central permitió reducir los ciclos productivos de 2 años y medio como promedio a 4 meses.

Caso 3, la implantación se extendió a todos los proyectos de la universidad lográndose cubrir con la implantación del 71,66% de las vistas del modelo. Como resultados positivo el modelo permitió asumir un proceso de adaptabilidad, en pleno inicio de la generalización (45 días transcurridos), han sido incorporado a este una vista y ocho nuevos elementos arquitectónicos estructurales a las vistas iniciales. Si se asumen los elementos de la nueva vista con los adicionados a las ya existentes se puede concluir que el modelo que se defiende en el presente trabajo ha iniciado un proceso de meta

transformación adaptativa en la organización que se generaliza, incorporando un total de 13 nuevos elementos, lo que representa un incremento del 21,66% de la versión original.

- El modelo ha constituido una guía metodológica y formal para la generalización y socialización del conocimiento en la comunidad científica de la universidad, especializada a la temática arquitectónica.
- El modelo constituye un marco de referencia horizontal que se centra en los conceptos y principios canónicos de la arquitectura y se abstrae de los procesos de desarrollo, gestión de proyecto, y de las tecnologías bases, lo que permite una comunicación horizontal entre los diferentes implicados en entornos productivos multitecnológicos, además de que multiplica su capacidad adaptativa.
- El modelo propuesto en los dos primeros casos de estudio presentó limitaciones en las capacidades conductuales y metodológicas que sirven de guía, problemas en la formalización de los escenarios arquitectónicos, en la formalización de las actividades metodológicas y la integración entre estas.
- Estos elementos fueron mejorados y el modelo está siendo desarrollado por el laboratorio de investigaciones en el paquete de gestión de proyecto GESPRO v1.0 (No. de registro CENDA Cuba paquete GESPRO v1.0 1540-2010).

Entre las Experiencias negativas del experimento se encuentra que el modelo resultante aplicado en la UCI presenta limitaciones en la formalización de la arquitectura, si bien constituye un instrumento metodológico y formativo importante, el mismo presenta limitaciones en la estructura formal para documentar la arquitectura.

Implantación e impacto en la socialización del conocimiento

Se evaluó el esfuerzo utilizado en los pasos de la metodología de implantación y la calidad y cantidad de los recursos empleados (próximos de 5 un mayor esfuerzo). Las celdas en blanco de la Tabla 2 representan que para el caso de estudio no se ha alcanzado estas etapas de la implantación.

Tabla 2. Datos resultantes de la aplicación de la metodología en los casos de estudio

	Metodología de implantación					Recursos involucrados	
	Diagnóstico	Planificación	Ejecución	Monitoreo y control	Mejora continua	Nivel Personal involucrado	Cantidad de Recursos
Caso 1	3	4	5	5	5	2	20
Caso 2	2	2	2	3	1	5	2
Caso 3	4	5			5	2	5

Partiendo de la Tabla 2 fueron calculados los índices de Esfuerzo Total en la implantación, que representa el trabajo total empleado para realizar los diferentes pasos en cada caso y cuyo valor está dado por la sumatoria del esfuerzo empleado en cada paso de la metodología según registro de la Tabla 2. Se calculó además el índice de rendimiento en la implantación, indicador que representa la eficiencia con que fue ejecutado el proceso de implantación, teniendo en cuenta el esfuerzo total, la cantidad de recursos y el tiempo empleado. En esta tabla se registra además el impacto en la socialización del conocimiento registrado en los casos de estudio ERP y SICS, valores que han sido normalizados a una escala de 0 a 1.

Tabla 3. Indicadores de la aplicación de la metodología en los casos de estudio

	Esfuerzo en la implantación	Tiempo empleado en la implantación	Índice de rendimiento en la implantación	Impacto en la socialización del conocimiento
Caso 1	24	8	0.15	0.80

Caso 2	15	4	1.88	1.00
Caso 3	16	1.5	2.13	



Figura 5 Comportamiento del rendimiento de la implantación respecto a la socialización del conocimiento.

Conclusiones

- Se identifican cinco corrientes teóricas fundamentales. Pero los modelos estudiados no integran todas las características necesarias para solucionar el problema que origina esta investigación y no integran las perspectivas contenidas en las corrientes teóricas identificadas.
- Se identifica la necesidad primaria de integración de las cinco escuelas fundamentales sinérgicamente y de formalizar la arquitectura usando las invariantes identificadas: Vistas, Reutilización, Procesos, Roles y Metas arquitectónicas
- El modelo arquitectónico de referencia basado en vistas, facilita el entendimiento y análisis de la arquitectura, promueve el trabajo en grupo y da soporte al aprendizaje desde la práctica. Permite además la especialización del arquitecto según áreas concretas de actuación.
- El modelo ha sido empleado con personal de diferente nivel educacional, preparación profesional y experiencia laboral (técnico medio, universitario y pre-grado) constatándose su impacto positivo en la socialización del conocimiento.
- En los casos de estudio analizados se constató el impacto del modelo en la competitividad evaluado en el aumento de la productividad, los ingresos por especialistas, la facilidad de mantenimiento, disminuyendo los tiempos de estabilización, los plazos de liberación (de años a meses) y los costos con amplia influencia de la reutilización.
- En la implantación del modelo se observa un período de estabilización: los primeros períodos de implantación (transición del Caso 1 al Caso 2) convergieron con la mayor cantidad de solicitudes de cambio en el modelo, aumentando el esfuerzo empleado. Al mismo tiempo los indicadores de rendimiento son los de mayor variación entre los diferentes escenarios de evaluación. Ver figura 5.
- Se observa maduración en el modelo durante la transición del Caso 2 al Caso 3 estabilizándose el rendimiento, evidenciando que el aumento del número de implantaciones, implica mayor madurez del modelo, apoyado en los conceptos de adaptación y mejora continua definidos en la metodología de implantación. Ver figura 5.
- En cada vista se caracteriza el escenario (condiciones específicas de funcionamiento, respuesta del sistema y características relevantes de su entorno) elementos que justifica ampliamente la decisión tomada evitando subjetividades.

- Se considera necesaria la introducción del modelo en los procesos de formación de los profesionales demandados por la industria del software. Se plantea introducir el modelo propuesto, en las asignaturas Modelos de Desarrollo de Software y Curso Básico de Gestión de Proyecto de la Maestría de Gestión de Proyectos Informáticos de la UCI, dado al impacto que tiene la arquitectura en la gestión de proyecto.

Bibliografía

Correspondencia

MSc. Michael González Jorrín. e-mail: michael@uci.cu

Laboratorio de Gestión de Proyectos. Dirección Técnica de la Producción.

Teléfonos: 53 7 837 2408 y 53 7 837 2429

Universidad de las Ciencias informáticas. Dirección: Carretera a San Antonio de los Baños Km 2 1/2 Torrens, Municipio Boyeros, La habana, Cuba. Página Web: <http://www.uci.cu/>