

SG[®]

Software Guru CONOCIMIENTO EN PRÁCTICA

Año 01 No.06 • Noviembre-Diciembre 2005 • www.softwareguru.com.mx

- Patrones de casos de uso
- Modelado conceptual
- Metodologías ágiles

[ENTREVISTA]

Javier Vega

Innovación en el S.A.T.

[ESPECIAL

Encuesta de salarios

Crecimiento Constante

Madurez de Procesos en Latinoamérica

**SI SE NECESITAN 18
MESES PARA
DESARROLLAR E
INTEGRAR UNA
NUEVA APLICACIÓN,**

ENTONCES YA NO ES NUEVA. ¿VERDAD?

Tu potencial. Nuestra pasión.™
Microsoft®

© 2005 Microsoft Corporation. Todos los derechos reservados. Microsoft, el logo de Windows, Windows Server System y "Tu potencial. Nuestra pasión." son marcas o marcas registradas de Microsoft Corporation en los Estados Unidos y/u otros países.

WINDOWS SERVER SYSTEM CON .NET LE AYUDA A CONSTRUIR E INTEGRAR MÁS RÁPIDAMENTE.

Su equipo de trabajo está entusiasmado con el nuevo sistema de la empresa. Les urge implantarlo y ver sus beneficios reales. Entonces, una enorme cantidad de código así como problemas de integración hacen parecer que se necesitará una eternidad para lograr que las cosas funcionen.

Windows Server System™ le puede ayudar. ¿Cómo? La clave es .NET.

.NET Framework, componente integral de Windows Server System, es el ambiente de desarrollo y ejecución que permite a diferentes aplicaciones trabajar en armonía, haciéndolas más fáciles de construir, manejar, implementar e integrar. .NET Framework emplea estándares de la industria como XML y Web Services que permiten que aplicaciones empresariales

de todo tipo se conecten a cualquier clase de infraestructura. Además, funciones como el mapeo automático de datos, hacia y desde XML, simplifican la integración, reduciendo la cantidad de código requerido para hacer que todo funcione.

Ya sea que implemente una solución ERP o bien que se encuentre unificando múltiples sistemas durante una fusión corporativa, si desea que su nueva aplicación todavía sea nueva al momento de implementarla, Windows Server System le puede ayudar. Descubra más de la integración de aplicaciones con Windows Server System y .NET en:

www.microsoft.com/mexico/windowsserver2003



MUCHAS COSAS PUEDEN PASAR EN 18 MESES.

Microsoft
**Windows
Server System**

■ A >

EDITORIAL



Nuestra región, Latinoamérica, vive un momento muy interesante en cuanto a la industria de Tecnología de Información, y especialmente la del software. Como sabemos, esta industria representa una gran oportunidad para nosotros. Sin embargo, para poder capitalizarla, debemos ser capaces de proveer productos y servicios de clase mundial. Es por esto que en los últimos años hemos visto un gran énfasis en incrementar la madurez de los procesos en las organizaciones de software. Hoy, estos esfuerzos ya comienzan a dar resultados, pero no por ello podemos relajarnos o pensar que ya lo logramos; al contrario, debemos redoblar el paso. No obstante, antes de esto, debemos darnos un tiempo para analizar nuestro recorrido, tanto lo que hemos hecho bien, como lo que hemos hecho mal, y actuar al respecto. También es necesario que compartamos experiencias y aprendamos unos de otros.

Todos los países en nuestra región tienen metas y obstáculos similares. Compartiendo podremos superarlos más rápido y con menos esfuerzo. Es bajo este contexto que hemos dedicado este número de SG a conocer los avances que llevamos, pero también a compartir la perspectiva de algunos de los consultores más experimentados y reconocidos en nuestra región.

En esta ocasión entrevistamos a Javier Vega, Director de Administración General de Comunicaciones y TI del SAT. Actualmente el sector gobierno esta inmerso en diferentes programas de mejora, y el SAT es muestra de los resultados que han tenido estos esfuerzos. Agradecemos a Javier Vega y a Tatiana Leonidovna por su tiempo y apoyo.

Adicionalmente, en este número presentamos los resultados de la primer encuesta de salarios SG. Antes de lanzarla, consideramos como positivo un escenario donde obtuviéramos 500 respuestas. El resultado fueron más de 1,000 respuestas de profesionistas, no sólo en México, sino también en Estados Unidos, Canadá, Argentina, Colombia, Guatemala y Venezuela. Muchas gracias a todos por su participación.

Agradecemos el entusiasmo de todos los colaboradores. Estamos muy entusiasmados por la gran respuesta que hemos tenido de su parte en diversos países. Les reiteramos que las puertas para colaboraciones editoriales están abiertas, pueden hacernos llegar sus propuestas, así como cualquier comentario o sugerencia a: editorial@softwareguru.com.mx

Equipo Editorial

■ DIRECTORIO

Edición Ejecutiva
Pedro Galván

Coordinación Editorial
Mara Ruvalcaba

Edición y Producción
Edgardo Domínguez

Dirección de Arte
Oscar Sámano

Fotografía
Gabriel González

Consejo Editorial
Francisco Camargo, Guillermo Rodríguez,
Ralf Eder y Raúl Trejo, ITESM CEM;
Hanna Oktaba, UNAM-AMCIS;
Luis Cuellar, Softtek;
Luis Vinicio León, e-Quallity - ITESO

Colaboradores
Ariel García, Jorge Palacios, Paulina Olivares,
Omar Ruvalcaba, Roberto Silva, Carlos
Pérez, Alejandro Bedini, Karin Steembecker,
Miroslav Pavlovic, José Guerrero,
Francisco Rivera, Irma García,
Atanacio Reyes, John Gómez, Saul Cuesta,
Luis Daniel Soto, Sergio Orozco,
Ernesto Corona.

Ventas
Claudia Perea

Marketing
Natalia Sánchez

Contacto
info@softwareguru.com.mx
+52 55 5239 5502

SG Software Guru es una publicación bimestral editada por Brainworx S.A. de C.V., Malinche no. 6, Col. El Parque, C.P. 53398, Naucalpan, México. Prohibida la reproducción total o parcial del contenido sin previo aviso por escrito de los editores. Todos los artículos son responsabilidad de sus propios autores y no necesariamente reflejan el punto de vista de la editorial. Reserva de Derechos al Uso Exclusivo: 04-2004-090212091400-102. Certificado de licitud de título: 12999. Certificado de licitud de contenido: 10572. ISSN: 1870-0888. Registro Postal: PP15-5106. Se imprimió en octubre de 2005 en Litográfica Roma. Distribuido por Sepomex.



EN PORTADA

Madurez de Procesos en Latinoamérica

La industria de software en Latinoamérica ha dedicado los últimos años a mejorar la madurez de sus procesos. Conozcámos algunos esfuerzos, resultados y perspectivas.

22

Productos

LO QUE VIENE 10

Lotus Notes y Domino, Flex, DBArtisan, Retrospect Backup,

A FONDO 12

GeneXus Yi

HERRAMIENTAS 14

SOA con BPMS

Columnas

Tejiendo Nuestra Red 06

por Hanna Oktaba

Mejora Continua 08

por Luis Cuellar

Cátedra y Más 32

por Irma García

Innovaciones en Software 43

por Luis Daniel Soto

Prueba de Software 46

por Luis Vinicio León

Prácticas

ARQUITECTURA 34

ebXML

En la segunda parte de este artículo, Atanacio Reyes nos lleva a través del escenario de ejecución de una transacción con ebXML

PROCESOS 38

Metodologías Ágiles

John Gómez comparte y explica los principios y valores de las denominadas Metodologías Ágiles

REQUERIMIENTOS 40

Patrones de Casos de Uso

Los patrones no solamente son útiles en el diseño de software. Saul Cuesta explica su aplicación en la gestión de requerimientos

SG[®]



Encuesta de Salarios 12

Estadísticas de salario en la industria



Entrevista 20

Javier Vega, Innovación en el SAT

En Cada Número

Noticias y Eventos	04
UML	44
Fundamentos	48
Tecnología	50
Biblioteca	52
Reflexiones	56

PDC 2005

Durante el pasado mes de Septiembre tuvimos la oportunidad de asistir al Microsoft Professional Developers Conference 2005 en Los Angeles, California. Dirigido principalmente a desarrolladores, este evento tuvo como objetivo dar a conocer el futuro de las herramientas y productos de Microsoft, como Windows Vista, Visual Studio 2005 y Office 12. Por medio de múltiples presentaciones, laboratorios y sesiones con expertos, concentraron a más de 6 mil desarrolladores provenientes de todo el mundo, que lograron visualizar el futuro de diferentes tecnologías, herramientas de desarrollo y medios digitales, entre otros.

Para conocer más detalles sobre el evento, visita el Blog de SG: softwareguru.com.mx o el sitio del PDC05: msdn.microsoft.com/events/pdc



Congreso AMITI 2005

El pasado 18 y 19 de octubre se llevó a cabo el 2do. Congreso AMITI 2005 "Las TI evolucionando los negocios", cuyo objetivo fue ofrecer un foro de ideas y propuestas para fomentar el crecimiento de las empresas mexicanas a través del uso de las Tecnologías de Información (TI). El evento logró reunir a los principales protagonistas de la industria de TI en los sectores de hardware, software, telecomunicaciones, consultoras, integradoras, proveedores y distribuidores, contando con la asistencia de los sectores público, privado y académico. El evento presentó una agenda bastante atractiva, cubriendo desde ponencias de figuras relevantes en el mundo de los negocios, una amplia área de exposición, pláticas de diversas empresas proveedoras de TI, hasta la entrega del Premio AMITI 2005 "Al aprovechamiento de las TI para el desarrollo de México". Vale la pena mencionar que el evento contó con una excelente organización y que el contenido de las diferentes ponencias y presentaciones fue de gran aprovechamiento para los asistentes.

Para mayor información, visita: www.amiti.org.mx

SAP Technology Day

Por primera vez en México, SAP presentó un evento dirigido a desarrolladores de software. El pasado 26 de octubre se llevó a cabo el SAP Technology Day, donde la empresa alemana anunció formalmente en México la disponibilidad del SAP Developer Network, de donde se puede obtener todas las herramientas necesarias para desarrollar aplicaciones para la plataforma Net Weaver.

Con esto, SAP busca incrementar el número de soluciones certificadas sobre su plataforma. Durante el evento se explicó la arquitectura de la plataforma Netweaver, y se llevaron a cabo demostraciones utilizando herramientas como el SAP Netweaver Developer Studio y el SAP Visual Composer.

El url para el SAP Developer Network es www.sdn.sap.com, y las presentaciones se pueden obtener en www.sap.com/mexico/company/events/technologyday/

IDC Directions 2005

El pasado 17 de octubre se llevó a cabo el 3er Foro IDC Directions 2005: "El valor real de las TI y su impacto en los negocios". Durante este evento IDC dio a conocer su visión sobre las tendencias y los temas clave del mercado de TI y Comunicaciones en México, como su comportamiento, predicciones específicas, y estrategias a futuro.

El Foro contó con ponencias de reconocidos analistas de la industria, diferentes talleres para canales, y un interesante panel donde se revisó el tema de la competitividad a través de las TI. Los diferentes análisis llevaron a cabo una introspección a los sectores financiero, de consumo, gobierno y comercio.

Los asistentes tuvieron la oportunidad de conocer el futuro de las TI y aprender como convertirse en verdaderos consultores de soluciones, más que proveedores de infraestructura.

Para mayor información visita: www.idc-eventos.com/directions05

Eventos

8 al 10 Noviembre 2005

Borland Developer Conference

Hotel Hilton, San Francisco, California

Info: www.borlandevents.com/devcon

Tel: 415 446 7709

e-mail: borlanddeveloperconference05@borlandevents.com

8 al 24 Noviembre 2005

DevDays Microsoft

8 Nov. Aguascalientes, 9 Nov. Cd. Del Carmen, 10 Nov. Puebla,

10 Nov. Mérida, 14 Nov. Veracruz, 22 Nov. Querétaro, 22 Nov.

Tijuana, 23 Nov. Mexicali, 24 Nov. Guadalajara

Info: www.microsoft.com/spanish/msdn/Mexico/eventos

9 Noviembre 2005

IV Encuentro GeneXus México

Hotel Nikko, Cd. de México

Info: www.genexus.com/mx/evento

Tel: +52 (55) 5255 4733

e-mail: info@genexus.com

9 al 11 Noviembre 2005

SEPG LA 2005 – 2da. Conferencia Latinoamericana del Software Engineering Process Group

Hotel Hilton, Guadalajara, México

Info: www.esi.es/SEPGLA

Tel: +52 (33) 3818 6778

e-mail: sepgla@esi.es

15 Noviembre 2005

Hacia una Industria de Software Competitiva – Apertura del Capítulo AMCIS Nuevo León

Centro Convex, Monterrey, Nuevo León

Info: www.amcis.org/nl/index.htm

Tel: +52 (81) 8625 2500

e-mail: nl@amcis.org.mx

18 Noviembre 2005

Mejorando los Servicios de TI para el Negocio y Automatizando la Admón de los Servicios de TI

Itera, Cd. de México

Info: www.itera.com.mx

Tel: (55) 5281 7670

e-mail: contactsalescenter@itera.com.mx

21 al 23 Noviembre 2005

6ta. Misión Expo de Software Español

Oficina Comercial Española, Cd. de México

Info: www.secartys.org/misiones.htm

Tel: +52 (55) 5281 2350 ext. 281

e-mail: exposecartys@yahoo.es

23 y 24 Noviembre 2005

Encuentro PROSOFT 2005 – Por una Industria de TI de clase mundial

World Trade Center, Cd. de México

Info: www.software.net.mx

Estudio Sobre Recursos Humanos en la Industria de Software

Secretaría de Economía dio a conocer un estudio realizado para determinar la cantidad y calidad de recursos humanos necesarios para el desarrollo de la Industria de Software en México. Dicho estudio fue realizado por la Universidad Autónoma Metropolitana a petición de la SE.

Dicho estudio tiene como objetivo central elaborar un diagnóstico de la oferta y la demanda actual de profesionales especializados en software, con el propósito de conocer la demanda futura y desarrollar escenarios prospectivos de las características que deberán tener los individuos y la industria mexicana de software en los próximos años.

El estudio reveló datos como:

- En el país existen 334,971 individuos laborando en departamentos de sistemas, de los cuales, 8 de cada 12 se desempeñan

en actividades relacionadas con software.

- El perfil promedio de los egresados es de hombres jóvenes entre 26 y 35 años.
- Existen 12,521 organizaciones que cuentan con más de 100 empleados.
- El 79.3% de las organizaciones son de tamaño mediano (100-1,000 empleados) y cuentan con menos de 10 empleados de sistemas en promedio.
- El estudio identifica una falta de alineación entre la oferta y la demanda de recursos humanos especializados en software. Las brechas más importantes tienen que ver con el título y el nivel académico.
- Para cumplir las metas del ProSoft se requiere mantener el crecimiento del mercado interno de software como producto y se necesita incrementar la tasa de crecimiento anual de 8% a 35%.

El estudio completo se localiza en:

www.software.net.mx

Recuerdos de Irlanda

CERVEZA, BAILE Y PROCESOS



La **Dra. Hanna Oktaba** es profesora en la Facultad de Ciencias de la UNAM. Es fundadora y vicepresidente de la Asociación Mexicana para la Calidad en la Ingeniería de Software. Actualmente dirige el proyecto con el cual se creó la norma mexicana para la industria de software.

A finales de agosto de 2005 el International Process Research Consortium (IPRC) tuvo su cuarta reunión en Dublín, Irlanda. Fue mi primera visita a este país y me sirvió para cambiar algunos prejuicios que tenía. Por ejemplo, no me gustaba la cerveza Guinness y descubrí que me gusta mucho; las chuletas de borrego tenían un olor que me molestaba y me di cuenta que las chuletas irlandesas tienen un olor muy agradable. También, tenía la idea de que México es el único país donde cualquier persona, bajo una modesta dosis de tequila o cerveza, expresa sus enormes dotes para cantar, bailar y divertirse, pero descubrí que lo mismo les pasa a los irlandeses quienes sustituyen la dosis de tequila por la de whisky.

Regresando al tema de la reunión de IPRC, ya fue la cuarta de las seis previstas y ya nos tocaba definir el esquema general del documento final sobre el mapa de líneas de investigación en procesos de software para los próximos diez años. Para que este mapa sea útil, se acordó describir “el punto de partida” y definir varios posibles destinos a alcanzar en el futuro. Los destinos se generarán a partir de diversos escenarios, que se han trabajado en la reunión anterior, y la descripción del estado actual sintetizará los elementos identificados en las primeras dos reuniones.

Para organizar el trabajo, que duró tres días, nos dividieron en dos grupos que llamaron *Push* y *Pull*. El objetivo del primer grupo es describir las fuerzas que empujan la investigación en procesos de software en ciertas direcciones. Mientras que el otro grupo identificará las fuerzas que “jalarán” la investigación a futuro, según el contexto de cada escenario destino.

El primer grupo, en el cual participé, tuvo como tarea revisar los documentos del trabajo previo y otros, que fueron proporcionados por los miembros del consorcio, para identificar tendencias y realidades en el desarrollo de sistemas. Llegamos a identificar 227 ideas que hay que clarificar, eliminar duplicidades y clasificar de tal manera que queden entendibles. Barry Boehm está coordinando los trabajos de este grupo y fue él quien nos propuso una primera aproximación de la estructura del contenido.

Uno de los momentos curiosos durante el encuentro fue la discusión sobre la definición del *proceso de software*. Desde las reuniones pasadas ya se había mencionado que no podemos proponernos definir las futuras líneas de investigación en procesos de software mientras no tengamos un acuerdo sobre este concepto. Para remediar el asunto, se delegó a dos de los miembros la labor de poner por escrito las posibles definiciones y su justificación. El documento que prepararon reveló que existen

por lo menos dos versiones antagónicas de la interpretación de lo que es un proceso. La escuela que yo llamaría SEI, opta por preocuparse en “qué hacer durante la realización de un proceso con el fin de obtener resultados deseados”, mientras que la escuela ISO/IEC15504 prefiere especificar “qué es lo que se propone lograr a través de un proceso”. Ambos enfoques son justificables y de alguna manera complementarios. Sin embargo, debido a que entre los miembros del consorcio hay “fundamentalistas” de ambas escuelas, la discusión final sobre este tema nuevamente se pospuso hasta la siguiente reunión.

Otro tema polémico surgió durante el panel con los representantes de la industria participantes en el consorcio. Trataron de llamar la atención del grupo académico sobre el enfoque del documento resultante de nuestro trabajo. Dijeron que éste no puede reducirse simplemente a una lista de temas de investigación “chidos” o “cool”, sino que debe de incluir una clara relación de estos temas con las necesidades de la industria y de la sociedad en general. Todo esto con la finalidad de que los organismos que otorgarán el financiamiento para la investigación en procesos de software tengan disponible una justificación convincente de los beneficios esperados.

En esta ocasión tuvimos un día extra de trabajo dedicado a un taller conjunto de IPRC con Irish Software Engineering Research Consortium. Este último agrupa representantes de la academia e industria de software local. Al escuchar su presentación me dio envidia de la buena. Sus objetivos son hacer investigación “amigable” para la industria local e internacional, y construir equipos de investigación y desarrollo de alta calidad. Para armonizar la investigación con las necesidades de la industria decidieron enfocar su atención en resolver los problemas de tres dominios específicos: sistemas automotrices, aparatos médicos y sistemas financieros. Y, además, como su país tiene la política de apoyo a la industria de software porque entienden que el software “tiene la fuerza de cambiar al mundo”, obtuvieron los recursos para convertir el consorcio en un Centro.

Finalmente, un representante de Australia presentó los argumentos por los cuales 275 organizaciones dedicadas al desarrollo de software de su país expresaron su falta de interés por adoptar CMMI. El más importante de los argumentos es que la relación costo-beneficio no está muy clara. El costo es muy alto, el tiempo de adopción muy largo y los beneficios a corto plazo dudosos. Prometo corroborar estos datos personalmente porque en noviembre tenemos la quinta reunión en Australia. ☺

- Hanna Oktaba

Evaluación de Productos de Software • Renta de Testers • Capacitación en Prueba de Software • Evaluación y Mejora de Procesos de Prueba • Puesta en Marcha de Organizaciones •



e•Quality^{Corp.}®
siempre • todo • mejor •

Nosotros probamos software...
¡Pónganos a prueba!

www.e-quality.net

Compartiendo Experiencias

AL MENOS, ¿APRENDEMOS DE NUESTROS ERRORES?



Luis R. Cuellar es Director de Calidad a nivel mundial de Softtek Information Services. Luis es reconocido por la American Society for Quality (ASQ) como Certified Quality Manager, Certified Software Engineer, y Six Sigma Black Belt. En los últimos cinco años ha estado a cargo de la definición e implantación de la estrategia para CMM5 y Six Sigma a través de las diferentes áreas del centro de desarrollo de Softtek.

Recientemente, durante una conferencia, unos colegas compartían su experiencia acerca de un proyecto que había fracasado. Durante la plática surgió la siguiente frase: “Por lo menos ya aprendimos cómo no hacer las cosas”. ¿Cuántas veces hemos escuchado esto? ¿Qué significa realmente? Según la Wikipedia, aprendizaje es “el proceso de adquirir conocimiento, habilidades, actitudes o valores a través de estudio, experiencia o intercambio de ideas; dicho proceso origina un cambio persistente, medible y específico en el comportamiento del individuo”.

Lecciones Aprendidas

¿Cómo generamos ese aprendizaje? Una práctica muy común al terminar un proyecto es llevar a cabo un análisis de lecciones aprendidas, para lograr que los individuos adquieran información que les ayude a tomar mejores decisiones en el futuro. Esta es una buena práctica, pero es limitada en cuanto a su alcance. Por un lado, típicamente se llevan a cabo al final del proyecto, cuando ya no se puede hacer mucho por éste. Por otro lado, los resultados son bastante subjetivos, y si una sesión de estas no es bien llevada, se convierten en un simple ejercicio para encontrar culpables (ya sean personas o circunstancias) de las fallas del proyecto. No me mal interpreten, éstas sesiones son útiles, pero su resultado es principalmente individual. Están orientadas a lograr que los participantes reflexionen sobre los elementos que afectaron negativamente al proyecto y qué hacer para evitarlos en el futuro. Desafortunadamente, no hace mucho para la organización como un todo.

Cambio Persistente

Entonces, ¿cómo aprende la organización? El aprendizaje indica un cambio persistente; analicemos el significado de esto. Según la primera ley de Newton, todo elemento mantiene su estado y dirección de movimiento mientras no se le aplique una fuerza externa. En nuestro contexto esto quiere decir que para generar un cambio se requiere de una fuerza directa que lo genere y mantenga. Desafortunadamente, por definición, toda fuerza tiene un costo. Así que para conseguir un cambio persistente, se requiere que el aprendizaje sea suficientemente claro y contundente como para generar un nuevo comportamiento y hacerlo parte de la cultura de la organización.

¿Cómo manejamos la claridad y contundencia del aprendizaje? En otras palabras, al terminar el proyecto que se barrió, ¿cómo sabemos qué fue lo que afectó la entrega? ¿Fue el líder? ¿Le faltaba experiencia? ¿La gente llegó tarde al proyecto? ¿Cuál de estas variables es el evento crítico? Si mañana tengo que iniciar un proyecto y sé que la gente no va a estar lista a tiempo, ¿detengo el proyecto?, o ¿iniciamos y después vemos cómo lo resolvemos? ¿No fueron esas las palabras que nos llevaron al fracaso en el proyecto anterior? Eso querría decir que el comportamiento no cambió en forma persistente, no hubo

suficiente fuerza aplicada para mover el comportamiento, por lo tanto, realmente no aprendimos.

Medible y Específico

¿Qué pasaría si pudiéramos comprobar que 100% de los proyectos que hemos hecho sobre cierta tecnología han tenido problemas y han implicado pérdidas para la empresa? ¿Estáramos más renuentes a iniciar un proyecto en esa tecnología? ¿Qué tanto sabemos sobre nuestros proyectos con problemas? ¿Cuántos proyectos barridos han tenido líderes novatos? ¿Cuántos han tenido reducciones de horas en su estimación inicial? La única forma de realmente crear la fuerza para cambiar el comportamiento son las mediciones que demuestren el hecho de que una acción nos lleva a una reacción. ¿Cuántos proyectos se han barrido? ¿Cuántas personas tenían? ¿En qué tecnología? ¿Quién era el líder? ¿Quién era el cliente? ¿Cuántos defectos tuvo? Entre más específicos seamos en la información, mejores decisiones podremos tomar, y esto nos llevará a un verdadero aprendizaje.

Listos, y Ahora, ¿qué Medimos?

Entonces como conclusión, el aprendizaje se da cuando tenemos datos que nos aseguren que entendemos exactamente lo que hicimos mal para no repetirlo. Pero, ¿cómo sé qué medir? La realidad es que no existe una respuesta única. Lo mejor que puedo hacer es responder con las siguientes preguntas: ¿qué quieren lograr? ¿Qué se está vendiendo? ¿Cuál es el diferenciador? De acuerdo a la teoría de Seis Sigma, una variable de salida se comporta como una función de una o varias variables de entrada. En otras palabras, $Y = f(X_1, \dots, X_n)$ por lo que para poder crear una organización de aprendizaje es necesario encontrar las variables de salida que buscamos y las variables de entrada que tienen correlación con estas variables de salida. Pero bueno, eso podría ser un libro aparte por lo que ya hablaremos más al respecto en el futuro.

Organizaciones de Aprendizaje

Finalmente, hablando de aprendizaje en base a intercambio de conocimiento, este mes decidí embarcarme en una nueva aventura, con la finalidad de crear una comunidad de aprendizaje que nos ayude a los individuos y organizaciones en países de habla hispana a cooperar y acelerar el proceso de aprendizaje sobre temas de calidad y software. El sitio es www.agentesdecambio.org y aprovecho este espacio para invitar a todos ustedes y las personas que conozcan interesados en calidad, mejora continua, Seis Sigma y demás, a participar, compartir sus experiencias y aprender en conjunto. Espero que este espacio nos sirva para asimilar más rápidamente el conocimiento y las prácticas necesarias para llevar a nuestras áreas de influencia a competir a nivel mundial. ☺

- Luis Cuellar

Por fin en México, el mejor CURSO de ¡GARANTIZADO por escrito!

Vea todo lo que incluye...



Le regalamos el examen de certificación de UML



El curso más práctico impartido por consultores expertos



Licencia GRATIS de Enterprise Architect para modelado en UML 2.0

GOLD+
MILESTONE CERTIFIED TRAINING



La primer empresa mexicana miembro de la OMG



Incluye el principal bestseller de UML a nivel mundial



GARANTÍA MILESTONE
Si no mejora sus proyectos le regalamos otro curso igual



Horas adicionales de asesoría para garantizar el aprendizaje

Expositor de



MILESTONE
CONSULTING

LOS ESPECIALISTAS EN UML, CMM Y ORIENTACIÓN A OBJETOS



* La OMG es el organismo mundial que define y controla UML.

info@milestone.com.mx

www.milestone.com.mx

Tel. +52 (55) 5684-4412



Macromedia Flex 2

DESARROLLO DE APLICACIONES RICAS
PARA INTERNET

Macromedia dio a conocer la línea de productos Flex 2, orientada al desarrollo de aplicaciones ricas e interactivas en Internet. La línea de productos está diseñada para cubrir un amplio espectro de escenarios, desde e-commerce hasta colaboración distribuida en tiempo real.

Los elementos que forman esta línea de productos son los siguientes:

- Flex Framework 2 – El modelo de programación y librería de componentes para Flex.
- Flex Builder 2 – Ambiente de desarrollo basado en Eclipse, con el que se desarro-

llan aplicaciones para el Flex Framework.

- Flex Enterprise Services 2 – El componente del lado del servidor que provee servicios de datos, soporte a pruebas automatizadas, e integración en arquitecturas orientadas a servicios.
- Flex Charting Components 2 – Componentes para visualización avanzada de datos.
- Flash Player 8.5 – La versión más reciente del tan conocido ambiente de ejecución en el cliente.

Las versiones alpha de Flex Builder 2 y Flash Player 8.5 ya están disponibles para su descarga. Se espera que estos productos esten disponibles comercialmente para la primera mitad del 2006.

Para mayor información, visitar:
www.macromedia.com/go/flex2intro



EMBARCADERO
TECHNOLOGIES.

Embarcadero DBArtisan 8.1

INCREMENTA SU SOPORTE DE PLATAFORMAS

Embarcadero Technologies anunció la disponibilidad de la versión 8.1 de su producto para administración de base de datos, DBArtisan. Esta versión ya incluye soporte para SQL Server 2005, en su edición CTP (Community Technology Preview). DBArtisan está disponible en tres ediciones diferentes:

- Estándar – Dirigido a quienes sólo necesitan administrar un conjunto reducido de bases de datos, y no requieren asistencia con almacenamiento, u optimización.
- Professional – Soporta un número ilimitado de bases de datos, e incluye depurador de SQL y analizador de SQL (*profiler*).
- Workbench – Además de las capacidades del Professional, esta edición incluye componentes para almacenamiento avanzado y gestión del desempeño.

Retrospect Backup

RESPALDO DIRIGIDO A PYMES



La empresa EMC lanzó la versión 7 de Retrospect Backup. Ésta incluye características nuevas y mejoradas que hacen aún más sencilla la copia de seguridad y la restauración para las pequeñas y medianas empresas (PyMEs), las empresas distribuidas, y los usuarios domésticos y profesionales.

Algunas de las características nuevas o mejoradas en Retrospect 7 son:

- Creación de copias de seguridad sintéticas sin afectar a los usuarios, aplicaciones ni a la red.
- Compatibilidad con dispositivos Fibre Channel e iSCSI.
- Soporte a las versiones más recientes de Red Hat Linux, SUSE LINUX y Mac OS X.
- Cifrado de datos AES de 128 bits en los medios de copia de seguridad.
- Compatibilidad con discos DVD de doble capa de 8 gigabytes.

* EMC regaló una licencia de este producto para ser rifada entre los subscriptores de Software Guru. Agradecemos a quienes participaron en la trivía y felicitamos al ganador, Fernando Sánchez Ramos, del Distrito Federal.

■ PRODUCTOS

Lotus Notes/Domino 7

MEJORAS PARA ADMINISTRADORES DE SISTEMAS

IBM lanzó formalmente en México Lotus Notes y Domino 7. Además de las mejoras en funcionalidad para el usuario, estos productos proveen nuevas capacidades y herramientas dirigidas a los administradores de sistemas y desarrolladores de software. Entre estas están:

- Mejor desempeño y escalabilidad. Domino 7 consume 25% menos recursos que la versión anterior, y es 80% más escalable.
- La administración es más sencilla, gracias al monitoreo de dominios domino, y la automatización de instalaciones en el cliente.
- Soporte a DB2 como almacenamiento de datos. La información se puede almacenar de manera indistinta tanto en la base de datos de Domino, como en DB2.
- Soporte mejorado a LDAP.

! Despierta i

Mientras trabajas otros disfrutan de
tu software o Desarrollo

La mejor solución antipiratería



Tlacoquemecatl # 21 - 501, Esq. Tejocotes Col. Del Valle México DF 03100

Tel: +52 (55) 5575-1441, 5575-1443/ 5575-1425 FAX: +52 (55) 5575-149

<http://www.safenet-inc.com> Contacto: rvargas@safenet-inc.com

GeneXus Yi

DESARROLLO ACELERADO

GeneXus Yi

ARTech

www.genexus.com

Calificación: 4 de 5



Precio y Disponibilidad

\$8,000 dls por licencia concurrente. Consultar el sitio de ARTech para encontrar un distribuidor local

Requerimientos de Sistema*

- Windows 98 o superior
- 64 MB de RAM (256 recomendado)
- Procesador tipo Pentium de 600 Mhz o superior
- .NET Framework 1.1

■ VEREDICTO

Conta con un conjunto de capacidades bastante rico. No es un producto barato, pero su precio se desquita fácilmente.

PROS:

1. Gran variedad de lenguajes.
2. Aplicaciones web con tecnología AJAX.
3. Patrones para reutilizar conocimiento de negocio.

CONTRAS:

1. Usabilidad del ambiente de desarrollo y su documentación.
2. Dificultad para leer e interpretar código generado.

GeneXus es una herramienta para desarrollar aplicaciones en múltiples lenguajes y plataformas sin necesidad de programar. En pocas palabras, es un generador de aplicaciones.

Desde hace décadas venimos oyendo sobre diferentes propuestas para generar aplicaciones sin necesidad de programar. Probablemente algunos de ustedes sean de los que piensan “es interesante, pero poco viable en la realidad”. Sin embargo, es innegable que cada día usamos más y más herramientas que generan código sin necesidad de teclearlo. Generar scripts SQL a partir de modeladores de datos es una forma de generación de código, generar deployment descriptors para J2EE desde un IDE, también lo es. Queramos o no, los días de “los verdaderos programadores usan VI”, cada vez están más lejanos, o por lo menos aplican a menos personas. Así que, bajo este contexto, hablemos de la nueva versión de este producto, llamado GeneXus Yi, que actualmente se encuentra disponible como *Release Candidate*.

Cómo Funciona

El paradigma de desarrollo en GeneXus se basa en la definición de entidades de información —denominadas transacciones— que maneja una aplicación. Al desarrollar una aplicación, básicamente lo que se hace es describir la visión del usuario final respecto a las transacciones que el sistema debe manejar.

El primer paso para desarrollar una aplicación es crear lo que se llama una “Base de Conocimiento” (*Knowledge Base*), o KB, para los cuates. Un KB es un repositorio que concentra el conocimiento necesario para generar una aplicación en múltiples plataformas. A su vez, un KB está compuesto por los siguientes modelos:

- Modelo de diseño – Contiene la definición de requerimientos, entidades y reglas. En sí, todo lo que es independiente de la implementación.

- Modelos de prototipo (uno o más) – Contienen la información específica a la implementación en una plataforma particular (por ejemplo WinForms con SQL Server).

- Modelos de producción (uno o más) – El mismo concepto que los modelo de prototipo, pero para cuando ya se desea liberar una aplicación.

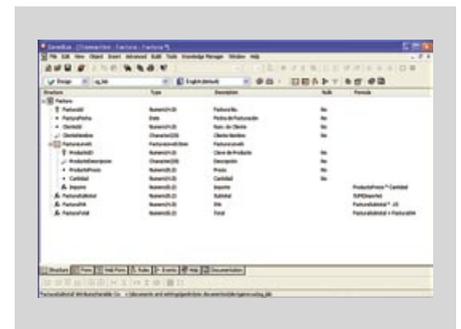


Figura 1. Definición de una transacción.

Aclarando el Mito

A estas alturas del partido, probablemente se estén haciendo la famosa pregunta: “¿Entonces con esta herramienta el usuario puede desarrollar sus aplicaciones él mismo y sin necesidad de personal de sistemas?”. La respuesta es... no. O por lo menos, muy difícilmente. En el escenario de desarrollo con GeneXus, hay un experto en la herramienta —denominado analista de negocio—, que captura en ella los requerimientos para generar la aplicación. Aunque el analista no requiere ser experto en programación, sí necesita tener una buena idea de cómo funciona una aplicación de software, así que típicamente será el personal de sistemas quien tome este rol. La diferencia es que en lugar de tener que estar al día en cuanto a los nuevos lenguajes y APIs, solamente requiere dominar GeneXus, y el resto del tiempo dedicarlo a entender mejor el negocio y generar aplicaciones mejor alineadas a éste. En resumen, es la misma idea de “preocúpate más por el negocio y menos por los bits” que venimos escuchando en todos lados.

La versión Yi incluye soporta las tecnologías más modernas como AJAX, Messaging EJBs y .NET Compact Framework.

Aunque como todos aquí sabemos, los bits tienen ese no sé qué que sé yo, que los hace tan adorables.

Plataformas Soportadas

Los lenguajes soportados incluyen los más comunes para aplicaciones en empresas, como Cobol, RPG, Visual Fox Pro, Visual Basic, Java y .NET. Esto incluye soporte a varias de las tecnologías recientes, como es el Compact Framework para dispositivos móviles en el caso de .NET, o EJBs (*Session y Messaging*) en el caso de J2EE. Entre los RDBMS tenemos a los sospechosos comunes: DB2, Oracle, SQL Server, y también opciones de código abierto como MySQL y PostgreSQL.

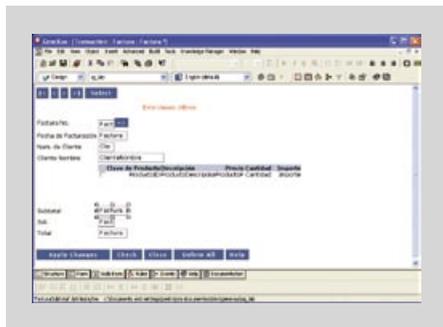


Figura 2. Forma web generada automáticamente para desplegar la transacción creada.

Nuevas Capacidades

La versión Yi provee diversas mejoras y nuevas capacidades. Entre las principales están:

- **Soporte a patrones.** La invasión de los patrones también ha llegado a GeneXus. En este caso, los patrones no están orientados a capturar y reutilizar diseño o código, sino estructura y comportamiento de objetos de negocio. Por ejemplo, uno de los patrones incluido es el de “Lista de Materiales”, con el cual se puede generar fácilmente una lista de los materiales que componen a un objeto de negocio. Obviamente, el usuario puede crear sus propios patrones y compartirlos.
- **Mejor experiencia de usuario.** Se han introducido una serie de detalles para mejorar

la usabilidad de las aplicaciones generadas, como tips inteligentes, mejor validación del lado del cliente, combo boxes dinámicos, etc. Adicionalmente, las aplicaciones web ahora incorporan tecnología AJAX para proveer una interfaz rica e interactiva.

- **Componentes de negocio.** Los componentes se pueden generar como web services o EJBs, de tal forma que puedan ser utilizados fuera del contexto de aplicaciones GeneXus

Lo que Podría Mejorar

Una preocupación común con este tipo de herramientas, es que las aplicaciones generadas tengan dependencia hacia el producto. En este caso, el código generado por GeneXus no provoca ésta dependencia para su ejecución. Las aplicaciones generadas no requieren componentes diferentes a si hubieran sido desarrolladas manualmente. Sin embargo, el problema está en que el código generado es poco entendible, por lo que sería difícil y poco práctico dar mantenimiento a la aplicación, sin utilizar GeneXus. Así que, en términos prácticos, aunque no hay una dependencia para la ejecución, sí la hay para el mantenimiento. Aunque esto no es completamente deseable, tampoco es tan malo. Sobre todo cuando se considera que una buena parte de los beneficios que otorga la herramienta —y que son los que realmente pueden ayudar a justificar la inversión— se dan precisamente cuando comienza el mantenimiento de las aplicaciones desarrolladas. Por ejemplo, si se tiene una aplicación generada para .NET 1.1, y en el futuro se decide migrar hacia la versión 2.0, la herramienta se encargará de ello. Así que después de todo, no es tan malo estar atado a ella.

La usabilidad del ambiente de desarrollo podría mejorar. Al parecer, ARTech se ha enfocado en que las aplicaciones generadas tengan una buena usabilidad, pero se ha olvidado un poco de la usabilidad para el desarrollador. Parece ser que conforme el producto ha ido evolucionando, ha ido perdiendo cohesión.

Por ejemplo, el componente para patrones se debe descargar y ejecutar por separado, y lo mismo pasa con el componente para ingeniería en reversa. Tal vez esto no hubiera sido un factor relevante hace unos años. Pero hoy en día, que en el mercado hay IDEs con gran énfasis en su usabilidad e integración transparente, esto resalta. La documentación parece sufrir el mismo problema. Está dispersa entre diferentes lugares, archivos y formatos. Espero que estos males sean sólo síntomas de que la versión Yi todavía es un *Release Candidate* y se corrijan cuando sea liberada la versión oficial.

Conclusión

En general, GeneXus es una herramienta bastante poderosa. Con ella se pueden generar aplicaciones sólidas y con gran experiencia para el usuario. Las empresas que requieren aplicaciones a la medida, pueden utilizar GeneXus para desarrollar éstas rápidamente y con un bajo costo de mantenimiento. Para las empresas que se dedican a hacer desarrollos para terceros, GeneXus les permite cubrir requerimientos aún sin contar con expertos en la tecnología específica. Otra oportunidad que se abre para estas empresas es la capacidad de reutilizar una base de conocimiento para generar aplicaciones para varios clientes.

En sus inicios, hace varios años, GeneXus era una herramienta de nicho, que solamente tenía sentido considerar en un número limitado de escenarios. Sin embargo, se ha mantenido en constante evolución, y hoy en día es una fuerte opción para la mayoría de las aplicaciones que desarrollamos. ☺

Agradecemos a Carlos Villafuerte y Carlos Álvarez de la empresa Imexsoft por su apoyo para generar este artículo. Su experiencia y comentarios fueron fundamentales.

SOA usando BPMS

IMPLEMENTACIÓN CON CARNOT Y CAI

Por Roberto Silva

En el entorno actual de cambios constantes, las organizaciones están obligadas a emprender una búsqueda continua de nuevos métodos para consolidar ventajas competitivas sostenibles. En 80% de los eventos de negocio que provocan cambios con impacto significativo, se requiere modificar código de alta complejidad, concentrado alrededor de la evaluación e implementación de reglas de negocio. Desafortunadamente, el mayor obstáculo para que las organizaciones se adapten ágilmente a cambios, son los ambientes heterogéneos generados por TI a lo largo de varios años, compuestos por islas automatizadas de sistemas inflexibles, que sólo generan beneficios marginales y locales a departamentos funcionales^[1].

Hoy en día, el enfoque tradicional de desarrollo y administración de software genera un desperdicio organizacional enorme y no permite responder adecuadamente a los nuevos retos, ya que los sistemas son contruidos para durar y resolver problemas funcionales de bajo nivel de abstracción. Para tener éxito, las organizaciones requieren un nuevo enfoque de solución, basado en el aprovechamiento de tecnologías de información diseñadas para soportar la adaptabilidad y la agilidad organizacional.

Nuevo Enfoque de Solución

El escenario delineado por estos hechos nos obliga a considerar un nuevo enfoque de solución a nivel proceso, arquitectura y

herramientas para el desarrollo de software empresarial; donde una de las metas centrales debe ser un manejo ágil y eficiente de reglas de negocio.

En este artículo, veremos un ejemplo de la plataforma y arquitectura del nuevo enfoque de solución, cuyos objetivos generales son: (a) Asegurar que la calidad de los sistemas se mantenga alineada a los objetivos cambiantes de la organización; (b) Contar con sistemas fácilmente adaptables a las necesidades de evolución de la organización; (c) Disminuir el riesgo, costo y complejidad de los proyectos de software; (d) Disminuir la complejidad de la administración del ciclo de vida completo de los sistemas (desarrollo, soporte y evolución).

BPMS

El nuevo enfoque está centrado en el uso de un BPMS. En este caso utilizaré como referencia el Carnot Process Engine, un producto de Carnot AG (www.carnot-usa.com). Este producto se basa en estándares abiertos, funciona sobre cualquier servidor de aplicaciones J2EE, y facilita la integración de aplicaciones y datos empresariales en ambientes heterogéneos bajo una arquitectura orientada a servicios (SOA). Para entender el funcionamiento de un BPMS como este, es importante entender cómo funciona el patrón Modelo-Vista-Controlador (MVC) ^[2].

En la Figura 1, podemos ver un proceso de adquisición de clientes, mapeado e integrado

en el CARNOT Definition Desktop, donde podemos ver aplicaciones y datos empresariales integrados a un mismo nivel de abstracción. El motor de procesos de CARNOT juega el rol de Controlador bajo el patrón MVC, ya que es el orquestador que sincroniza a todas las entidades de negocio requeridas para generar valor. En este caso, otorgando un crédito a un nuevo cliente. En el mismo contexto, las aplicaciones (“Página Vista de Promoción”, “Página de Actualización” y “Panel de Autorización”) son las Vistas y los datos (“Crédito” y “Clientes”) son los Modelos.

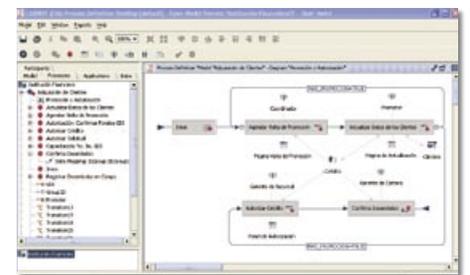


Figura 1. Aplicaciones y Datos Empresariales Integrados en el CARNOT Definition Desktop.

Interfaz Rica de Usuario

Otro aspecto clave de esta solución es contar con un elemento para generar interfaces de usuario ricas de manera rápida y sencilla. En este caso, mi voto se va al Composite Application Integrator (CAI) de Software AG (www.softwareag.com), que es una herramienta para desarrollo acelerado de RIAs (Rich Internet Application) utilizando AJAX.

Roberto Silva es VP de Marketing y Desarrollo Organizacional de TCDS (www.tcds.com.mx). Es consultor asociado de FUNDInnova y miembro activo del BPMG.org, BPMInstitute.org y KMInstitute.org. Estudió Actuaría (UNAM), cuenta con diploma y certificaciones en 8 Omega (BPMG), dirige el desarrollo permanente de los Diplomados Impulsare, cuenta con certificaciones de Rational en Implementación de RUP, Análisis y Diseño Orientado a Objetos usando UML, Administración de Requerimientos usando Casos de Uso, y Administración de Proyectos de Software Iterativos.

Encuesta de Salarios 2005

CUÁNTO, DÓNDE Y PORQUÉ

¿Sabes si la compensación que recibes es justa? ¿Crees que podrías ganar más en otro lugar? ¿Quieres saber qué área de especialización es la que más te conviene? ¿Vale la pena estudiar una maestría? Estas y otras preguntas nos hacemos a diario los profesionistas de sistemas. Por ello decidimos realizar una encuesta de salarios. La encuesta fue contestada por más de mil profesionistas de sistemas – 1,043 para ser exactos – a través de nuestro sitio web durante los meses de septiembre y octubre del 2005.

¿Cuánto Gana un Profesionista de Sistemas en México?

La siguiente tabla indica los tabuladores de salario que se manejaron en la encuesta, así como el porcentaje de personas que cae en cada uno.

Rango de salario	Pct.
Menos de 3 mil pesos	4%
De 3 a 8 mil pesos	17%
De 8 a 15 mil pesos	31%
De 15 a 25 mil pesos	31%
De 25 a 40 mil pesos	14%
De 40 a 60 mil pesos	5%
De 60 a 80 mil pesos	1%
Más de 80 mil pesos	1%

Tabla 1. Tabuladores de salario

Estos números arrojan un promedio ponderado de \$18,469 pesos

De acuerdo a la encuesta, esta cantidad es el salario bruto mensual de un profesionista de sistemas en México. Tengamos presente este número, ya que a continuación analizaremos algunos de los factores que pueden afectarlo.

Esquema

Uno de los primeros dilemas a resolver es el esquema de pago. Realmente quien decide esto es la empresa, no el empleado, pero aún así es bueno estar informado sobre qué es más conveniente. La siguiente tabla muestra los salarios promedio dependiendo del esquema de pago. Se incluyen dos cantidades, la primera es el sueldo bruto mensual, y la siguiente es un salario integrado, que contabiliza las prestaciones e incentivos monetarios.

Esquema	Pct.	Bruto	Integrado
Honorarios	25%	\$18,692	\$20,150
Nomina	67%	\$18,650	\$21,013
Independiente	9%	\$16,778	\$18,354

Tabla 2. Esquema de pago

Como podemos ver en estos resultados, el salario bruto por honorarios es mayor, pero por una diferencia mínima. En cambio, el salario integrado por nómina es significativamente mayor que en honorarios. En el caso de los independientes, la compensación es menor en ambos casos.

Dime Dónde Vives y te Diré Cuánto Ganas

Tanto la demanda de profesionistas de TI, como el costo de la vida, varían substancialmente dependiendo del lugar del país. Los estados de la república de donde hubo mayor respuesta, y que en general es donde hay más profesionistas de TI, fueron el Distrito Federal, México, Nuevo León y Jalisco. El resto de los estados los agrupamos como región Norte o Sur. La siguiente figura muestra la distribución por regiones, y el salario promedio en cada una.

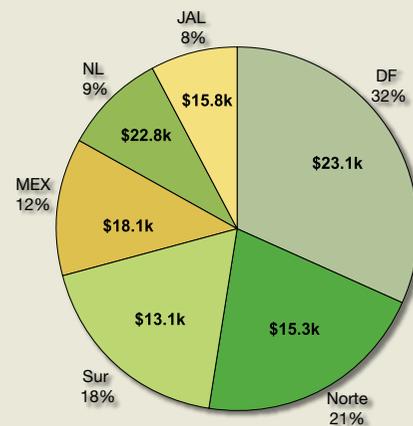


Figura 1. Distribución geográfica de la muestra.

Podemos ver que los estados mejor pagados son el D.F. y Nuevo León. Sin embargo, hay que recordar que el costo de vida en estas entidades también es significativamente más alto que en otras regiones del país.

¿Proveedor o Consumidor?

Que conviene más, ¿trabajar para las empresas que proveen productos/servicios de TI, o para el departamento de sistemas de una empresa que los consume?

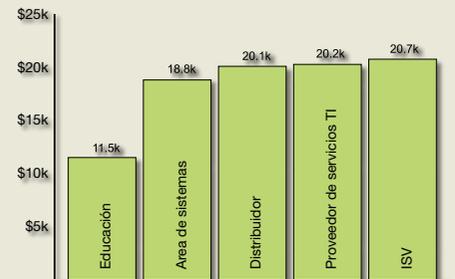


Figura 2. Salario promedio por tipo de organización.

Al parecer, la paga como proveedor es mejor que como consumidor.

Algo que también queda muy claro de esta gráfica, es que los sueldos en las instituciones educativas están muy por debajo de los demás. Definitivamente para tener mejores profesionistas necesitamos mejores maestros, y para tener mejores maestros se necesitan mejores sueldos. Ojalá esta tendencia pueda ser revertida.

Función

La función desempeñada en el trabajo obviamente es un factor crucial en la compensación. La figura 3 ilustra algunas de las funciones más comunes, con su salario promedio.

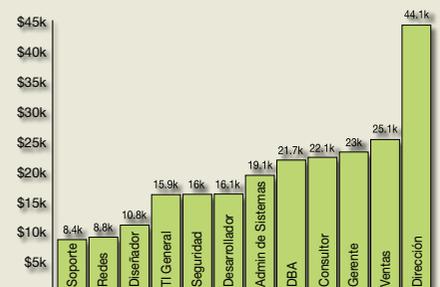


Figura 3. Salario promedio por función.

Demografía

Nuestra industria refleja una diferencial de cerca del 20% entre el sueldo promedio de una mujer (\$15,692) y el de un hombre (\$18,796). La edad también es otro factor importante, como lo refleja la figura 4. Conforme ésta es mayor, aumenta la experiencia

y por lo tanto el sueldo. Es importante notar que a partir de los 50 años ya se nota un descenso en la compensación.



Figura 4. Sueldo promedio por rango de edad.

Más allá del sexo o la edad, el factor demográfico de mayor relevancia en nuestra industria es el máximo grado de estudios alcanzado. Esto tiene sentido ya que nuestra profesión es altamente intensiva en conocimiento.

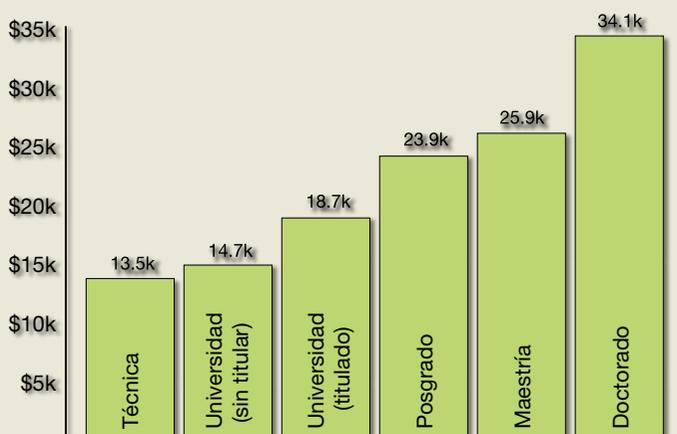


Figura 5. Sueldo promedio por grado de estudios.

¿Antigüedad o Experiencia?

¿Que se paga mejor, la experiencia en el oficio o la antigüedad en la organización?

Afortunadamente la experiencia tiene mayor peso que la antigüedad. Esto nos da mayor flexibilidad para cambiarnos de organización, sin arrepentirnos por la antigüedad que vayamos a perder.

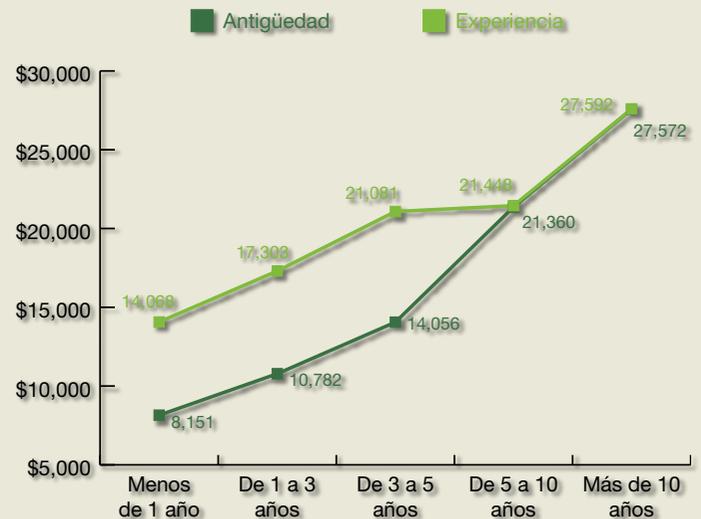


Figura 6. Sueldo promedio por antigüedad y experiencia.

Conocimientos y Habilidades

Como sabemos, nuestra profesión cuenta con una gran variedad de tecnologías y áreas de conocimiento en las que uno se puede especializar. ¿Qué lenguajes de programación son los más cotizados? ¿Cuales son las certificaciones mejor pagadas? Veamos los resultados



Figura 7a. Lenguajes de programación.

Algo que podemos apreciar es que las tecnologías open source son las que significan los salarios más bajos. Esto es señal de la baja penetración de estas tecnologías en los segmentos de mayor valor, al menos en nuestro país.

Llama la atención que Cobol sea el lenguaje mejor pagado. ¿Qué no era cosa del pasado? Bueno, lo que sucede es que las personas que

saben Cobol, son las de mayor experiencia, y como ya vimos, la experiencia es bien remunerada.

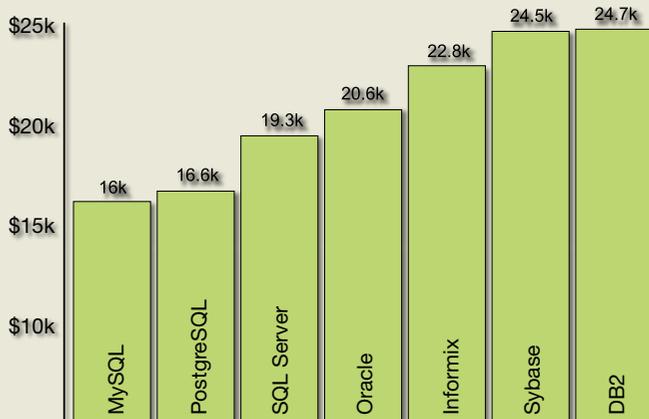


Figura 7b. Bases de datos.

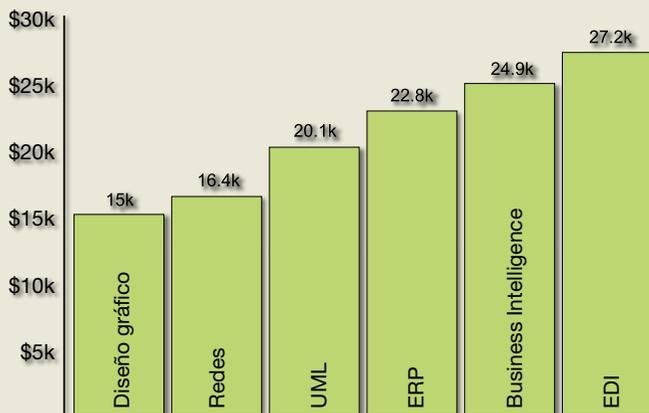


Figura 7c. Otras habilidades.

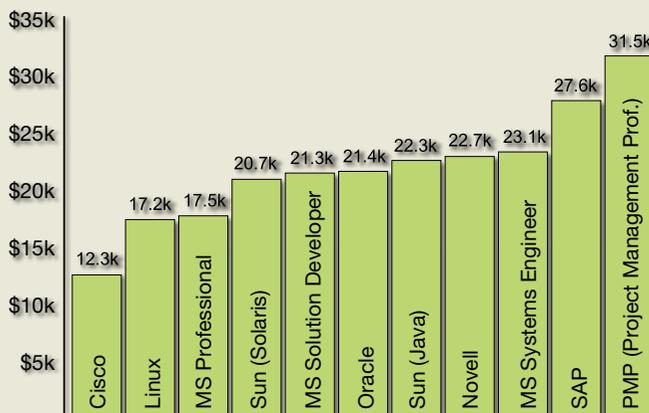


Figura 7d. Certificaciones.

Variación en los Últimos 12 Meses

¿Que tanto han crecido los salarios de los profesionistas de sistemas en el último año? La gran mayoría de los encuestados reportó que su salario no cambió, o aumentó menos de un 5%, lo cual indica simplemente un ajuste de acuerdo a la inflación. Los detalles son los siguientes:

Variación en salario

Disminuyó	6.9%
Sin cambio	40.5%
De 1 a 5%	24.6%
De 6 a 10%	13.7%
De 11 a 20%	6.8%
De 21 a 30%	3.5%
De 31 a 50%	2.1%
De 51 a 100%	1.6%
Más de 100%	0.4%

Prestaciones

Al analizar un esquema de compensación, no podemos dejar de revisar las prestaciones ofrecidas. La siguiente tabla indica las prestaciones más comunes, y el porcentaje de los encuestados que recibe cada una.

Prestaciones

Gastos médicos mayores	39%
Horario flexible	37%
Préstamos	33%
Bonos por desempeño	27%
Gastos médicos menores	17%
Teléfono celular	14%
Ayuda para educación	12%
Trabajo desde casa	11%
Automóvil	6%
Ayuda familiar	6%
Departamento	6%
Acciones (stock options)	5%

Satisfacción y Motivación para Buscar Trabajo

En general, los encuestados están más satisfechos con su trabajo que con su compensación. Esto no es ninguna novedad.

Grado de satisfacción	Con el trabajo	Con la compensación
Muy satisfecho	43.2%	12.4%
Algo satisfecho	45.3%	52.1%
Poco insatisfecho	8.3%	23.6%
Muy insatisfecho	3.2%	11.9%

En cuanto al estatus de búsqueda de trabajo, tenemos 4 grupos: quienes activamente buscan trabajo, quienes considerarían una mejor oferta (pasivos), quienes están buscando una mejor posición en la misma empresa, y quienes no están buscando. La distribución por grupo se ve de la siguiente manera:

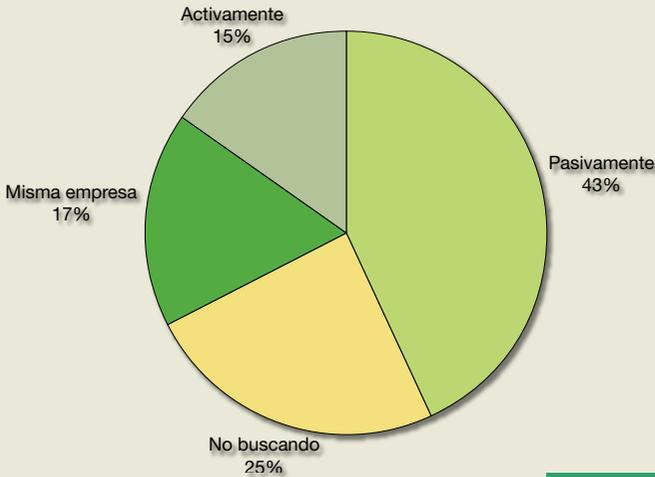


Figura 8. Estatus de búsqueda de empleo.

Analizando las razones por las que cambiarían de trabajo, tenemos lo siguiente:

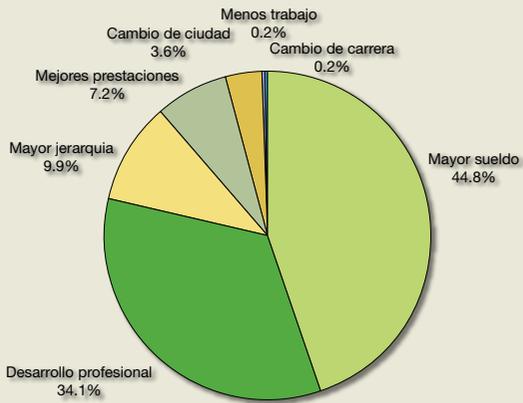


Figura 9. Razones para cambiar de empleo.

Conclusión
 Hemos visto los salarios promedio para profesionistas de sistemas en nuestro país, así como algunos de los principales factores que lo afectan. Si tú crees que la compensación que recibes no es justa, trata de hacer algo al respecto. Si es justa pero no te satisface, debes desarrollarte profesionalmente. Aquí te hemos dado algunas ideas de caminos que puedes tomar.
 Por último pero no por ello menos importante, agradeceremos enormemente la participación de todos los que llenaron la encuesta. Esperamos que la información generada les sea de utilidad.

Transformamos la Calidad en TI

Servicios de Consultoría, Capacitación y Evaluación, con experiencia y reconocimiento a nivel mundial en la implantación de prácticas innovadoras de Administración de Proyectos e Ingeniería de Software para la Industria de Tecnologías de Información.

www.avantare.com
 Tel. +52(55) 55-44-33-21

CMMI®
 PMBOK®
 ISO9000
 CMMISM

JAVIER VEGA FLORES

DIRECTOR DE ADMINISTRACIÓN GENERAL DE COMUNICACIONES Y TECNOLOGÍA DE LA INFORMACIÓN DEL SAT

Desde diciembre de 2003, el Ing. Javier Vega Flores encabeza la Dirección de Administración General de Comunicaciones y Tecnología de Información, del Servicio de Administración Tributaria (SAT), dependiente de la Secretaría de Hacienda. Bajo su administración, esta dependencia ha tenido avances importantes en cuestión de tecnología e innovación.

¿Cómo se integra a dirigir esta área?

Mis orígenes son de iniciativa privada, tengo un perfil inicialmente orientado a temas administrativos y financieros, y posteriormente orientado a planeación estratégica. El objetivo al integrarme al sector público era poder aprovechar mis antecedentes en planeación estratégica, para apoyar el proceso de cambio que buscaba implantar el SAT.

¿Qué contraste al llegar a esta área?

Encontré un equipo muy comprometido, con gran disposición por alcanzar la excelencia, y con claros intereses de mejorar los niveles de servicio al usuario. Pero también encontré una organización dispersa, con diferentes organizaciones independientes de tecnologías de información, donde estaba un poco complicado alcanzar un nivel de integración.

¿Cuál es el estado de esa reingeniería y la evolución hacia la visión planteada?

Siempre hay una posibilidad de mejora. Aquí el objetivo no es hacer el "método Vega" y que mañana venga el "método González" y lo cambie. El objetivo es tomar criterios preestablecidos aceptados universalmente y tratar de madurar sobre ellos. Por eso escogimos estrategias probadas como CMMI, PMI, ITIL y COBIT. El objetivo es que todo lo que hagamos sea de manera integrada para que sea algo permanente.

¿Cómo convencieron a los involucrados de que adopten las mejoras?

Tuvimos varias sesiones, en las que el Secretario mismo reunió a todo su staff, hasta niveles de Subsecretarios y Directores de Área, para explicar la importancia del proceso de cambio y hacia donde queríamos llegar. No se requirió mucho esfuerzo para convencer, ya que había suficiente convencimiento interno de la necesidad de aplicar mejores prácticas.

¿Se apoyaron en gente externa o únicamente en personal interno?

Realizamos una mezcla. Yo creo que un proceso de cambio es imposible si tu gente no lo está haciendo convencida, así que ellos fueron los líderes del cambio. Lo segundo fue allegarnos

de consultoría para la reingeniería de procesos. Formamos un equipo interno que actuó como "sombra" de los consultores y posteriormente tomó su lugar.

¿Cómo se justifica el iniciar y mantener este tipo de proyectos?

Le avisamos a las cabezas de los usuarios que iban a tener implicaciones y que no se sorprendieran de ver algunos problemas en la implantación. También vendimos los éxitos logrados para sembrar en la gente la confianza de que era posible llevar a cabo el cambio. Los niveles de madurez de procesos no los vas a conseguir al día siguiente, ya que por muy buena tarea que hagas en el diseño de procesos, la implementación implica también un proceso de madurez en sí.

¿Qué ha sido lo más difícil?

El no haber tenido el tiempo suficiente para hacer una implementación que implica logros de mediano y largo plazo. Sin embargo, tomé la decisión de arrancar ese proceso de cambio. Para mí lo más importante es poder dar la satisfacción a la gente de alcanzar los reconocimientos, y darles el sentimiento de propiedad. Hemos recibido 3 reconocimientos del premio Innova, tenemos la certificación ISO 9000 en la Administración de Laboratorio, totalmente alineada a ITIL, y esperamos obtener CMMI3 para mediados del 2006.

¿Cuál es tu balance de lo que cuesta un programa de mejora contra lo que vale?

Sí entendemos el impacto que puede tener un anuncio de televisión comparado contra el impacto que puede tener una implementación de procesos en tecnologías de la información o en nuevas prácticas, resulta mucho más fácil decir. Tienes que hablar el lenguaje del Comité Directivo, para que puedas hacerles comprender los efectos esperados. El costo elevado es para la gente, porque no les vas a quitar el trabajo del día a día y la gente va trabajar por las noches, los fines de semana, "la tarea arriba de la tarea". Este costo se nos olvida. Por otra parte, no puedes permitir y tampoco puedes esperar que llegue un tercero a implementarlo todo por ti.

¿Cómo miden el retorno de inversión en un proyecto como este?

Existe un comité de proyectos que evalúa las iniciativas. Cada iniciativa tiene que pasar por un periodo de caso de negocio. Aquí se establecen los beneficios esperados, como podría ser un aumento en la recaudación, u otros que tienen que ver con oportunidades de soporte y servicio. Tal es el caso de la seguridad informática requerida por quienes pagan sus impuestos vía Internet. Otro ejemplo es el de mejoras en la operación. Si vas a una oficina local de recaudación y te vas a tardar 45 minutos en pagar tus impuestos contra 5 minutos que te vas a tardar al hacerlo por Internet, la pregunta de cuál va a ser el retorno de la inversión sobre eso, queda resuelta.

¿Cómo ha afectado esto la relación con tus proveedores?

Lo que buscamos es enfocarnos en nuestro negocio y no enfocarnos a temas que son de soporte operativo. Hemos centralizado aplicaciones que estaban distribuidas, incluyendo un centro de datos para elevar sus niveles de servicio mediante la asignación a una compañía especializada. Hoy en día tenemos una gran cantidad de contratos en licitación, que están orientados a todos los procesos de infraestructura, tanto en lo que podían ser equipos de escritorio, como redes LAN, telecomunicaciones, centro de datos, almacenamiento y procesamiento de datos bajo demanda. En estas iniciativas lo que nos importa no tanto es definir la tecnología, sino la definición de acuerdos de nivel de servicio.

Algunos proveedores indican que al aumentar los requisitos de entrada, empiezan a calificar menos proveedores nacionales. Incluso si se suben lo suficiente, no califica ninguno, ¿qué les dirías?

Lo que se busca son las alianzas entre ellos. Las organizaciones grandes, como bancos u organizaciones públicas, requieren de tener una contraparte que sea igualmente robusta. Sabemos que muchos proveedores grandes son en realidad grandes integradores, y

así los pequeños aprenden prácticas que de otra manera no hubieran alcanzado. Hemos tenido excelentes experiencias con proveedores totalmente mexicanos que nos han cuidado en muchos de los procesos, incluso a nivel de revisión de proceso.

¿Qué sigue en la ruta de mejora para las TI en el SAT?

Estamos pretendiendo la continuación de la primera etapa tanto por la parte de procesos de operación y en la parte de soluciones de negocio y de desarrollo. Tenemos pilotos que nos deben garantizar un cierto nivel de madurez en el resto de la organización. Al final de la administración, esperamos que la gente que trabaja en TI tenga un gran entendimiento del negocio, y que el servicio que preste esté basado en un acuerdo de servicio del negocio.

¿Qué beneficios plausibles puede esperar un importador, exportador o un contribuyente?

En las aduanas, aproximadamente se están cruzando cerca de 450 mil millones de dólares entre importación y exportación anual. Todas las importaciones pasan por un proceso de "desaduanamiento", esto implica que, de no haber contado con este proceso de cambio, hubiera complicado el comercio en estos niveles. Contamos con producciones "justo a tiempo" para bienes de producción. En el lado de servicios al cliente, hoy pueden hacer una cantidad de operaciones de manera virtual. Este año la gente pudo percibir que, si cumplía con ciertos criterios, una devolución se realizaba en los tiempos comprometidos con el Congreso.

¿Tienen planeado implementar MoProSoft?

Yo creo que MoProSoft es un esfuerzo importante para el gobierno mexicano, pero lo que tenemos que recordar es el nivel de madurez. En ese contexto, yo creo que buscar la certificación de CMMI nivel 3 es un paso todavía por encima de MoProSoft. Lo que yo pediría a mis proveedores es que igualaran, o de ser posible superaran los niveles de madurez en mi organización.

¿Qué le dirías a los directivos de áreas de TI del sector público que no han iniciado un programa de mejora?

Los invito a que den el primer paso aunque estén en el último año de gobierno. No es para dar el resultado final, sino para iniciar una práctica que pueda tener una continuidad hacia delante, y mientras sea una practica aceptada universalmente, no veo ninguna razón para que ésta sea suspendida. ©



»» CRECIMIENTO CONTINUO

Estudio Sobre la Mejora de Procesos en México

Por Carlos Pérez

Los programas de mejora de procesos de desarrollo de software en México son relativamente nuevos. Hasta hace algunos años el éxito de los proyectos se fundamentaba principalmente en la capacidad de sus directivos y en el personal que laboraba en los proyectos de las diferentes organizaciones. En la actualidad son los propios directivos quienes se dan cuenta que esta capacidad no es suficiente y que es necesario definir procesos y establecer la disciplina que permita repetir y reutilizar las experiencias obtenidas de proyectos exitosos. En menos de diez años este enfoque ha tomado presencia dentro de las organizaciones de desarrollo de software y áreas internas de TI. Cada vez es más demandado y reconocido el éxito de los proyectos respaldados por un enfoque de procesos. La capacidad que se logra permite llevar de un desarrollo artesanal a un desarrollo industrial consolidado. Pero aún queda mucho por hacer, el alcance de estos programas no rebasa 5% del total de las organizaciones de desarrollo de software en México.

Existen diferentes esfuerzos para llevar la cultura de procesos a nuestras organizaciones. Una muestra de este esfuerzo de difusión y concientización dentro de la industria fue la creación de la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS) y el lanzamiento por segundo año del SEPG LA en México. El Programa para el Desarrollo de la Industria del Software (ProSoft) como proyecto de Gobierno ha permitido acelerar la transformación hacia una industria de desarrollo de software orientada a procesos, principalmente por los trabajos de la mesa 6 que se enfoca en alcanzar niveles internacionales en capacidad de procesos. El establecimiento de la Norma Mexicana para el desarrollo y mantenimiento de software (MoProSoft) y los proyectos relacionados, establecen las bases para facilitar ese trabajo. Por otra parte, los requerimientos de los mercados internacionales y los proyectos a escala mundial, han hecho reflexionar a muchos directivos sobre la necesidad de iniciar programas de mejora de procesos y mejorar su competitividad y desempeño actual.

En diversos foros y eventos se han divulgado y difundido los resultados alcanzados por las organizaciones de desarrollo de software en el país. Pero en muchas ocasiones presentan un enfoque aislado, no concentran toda la información o no muestran las interrelaciones que se tienen y lo que esto representa dentro de la industria. Esta situación provoca, en muchos casos, el desconocimiento del alcance y dimensiones que se ha logrado en estos programas y cuáles son las lecciones aprendidas para los que se inician en la “aventura”. Por otra parte, en numerosas ocasiones los clientes nos han solicitado información al respecto y para ello hemos recopilado información

que nos permita ofrecerles los datos que requieren, aún cuando no siempre contamos con la fuente oficial.

Considerando estos elementos nos dimos a la tarea de realizar un estudio más detallado sobre la mejora de procesos en México, dado principalmente por la carencia de métricas al respecto. Básicamente nos apoyamos en encuestas y recopilación de información de las organizaciones involucradas de acuerdo con nuestra experiencia y conocimiento de la industria. En muchas ocasiones influyen cuestiones de confidencialidad, estrategias de mercado y otros elementos que dificultan obtener la información completa. Se orientó hacia las organizaciones que han implementado programas de mejora basados en el Modelo de Madurez y Capacidad (CMM) y se indican otros modelos y normas utilizados.

El propósito principal de este estudio es proporcionar a las organizaciones de desarrollo de software una fuente de información que les dé la posibilidad de compararse con otras organizaciones, así como para servir de base para estudios posteriores. Los resultados permiten conocer el nivel alcanzado por estas organizaciones, y características que las permitan clasificar dentro del mercado, como: tamaño de la organización, sectores que atiende, tiempo dedicado al proyecto, resultados obtenidos, modelos de referencia utilizados, entre otros. De igual forma se presenta un análisis sobre las necesidades, barreras y beneficios que reportan estas organizaciones al iniciar un programa de mejora de procesos. Como consecuencia, se puede conocer la situación actual del mercado en cuanto a la mejora de procesos de software

y lo que representa para el fortalecimiento de la industria en México.

La encuesta se aplicó a tres grupos principales de interés, considerando:

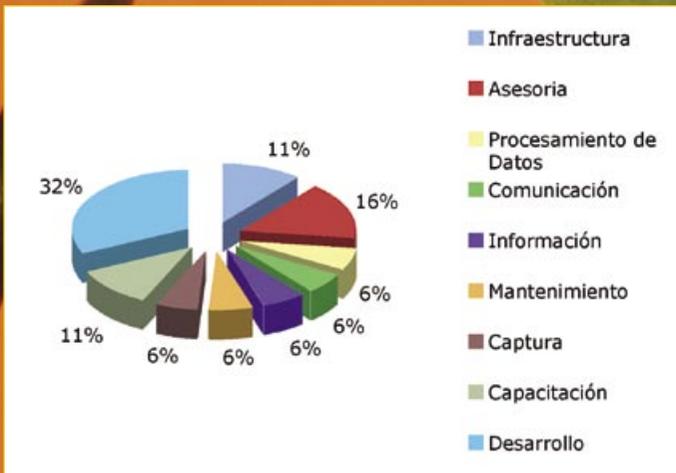
1. Organizaciones que han sido evaluadas y que son identificadas en el mercado.
2. Organizaciones que están trabajando en un programa de mejora.
3. Organizaciones interesadas en implantar o iniciar mejoras de sus procesos actuales.

Las preguntas se hicieron en términos de conocer:

- Segmento de mercado, actividad económica y servicios informáticos que ofrece cada organización.
- Tamaño, antigüedad en operación y distribución geográfica de las oficinas.
- Duración promedio y tamaño de los equipos de proyectos.
- Modelos de referencia que se utilizan, así como las evaluaciones y resultados alcanzados.
- Necesidades, barreras y beneficios obtenidos.

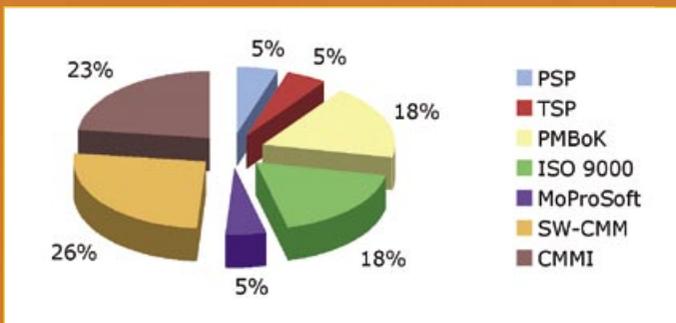
De acuerdo con la información obtenida, el perfil general de las organizaciones que inician un programa de mejora refleja que tienen ocho años de operación en el mercado, orientadas principalmente a los servicios profesionales y de mantenimiento, con actividad económica enfocada hacia el área de servicios y dedicados principalmente al desarrollo de sistemas. Por otra parte, la duración de los proyectos es de seis meses a un año, con equipos de proyecto de dos a cinco personas y un tamaño pequeño de Organización, con menos de 100 personas. Se alcanza normalmente el nivel 3 de madurez, utilizando como modelos de referencia CMM y PMBoK, y los ciclos de mejora tienen una duración de once meses.

La siguiente gráfica refleja el tipo de servicios que ofrecen las empresas consultadas:



Gráfica 1. Distribución por tipo de actividad.

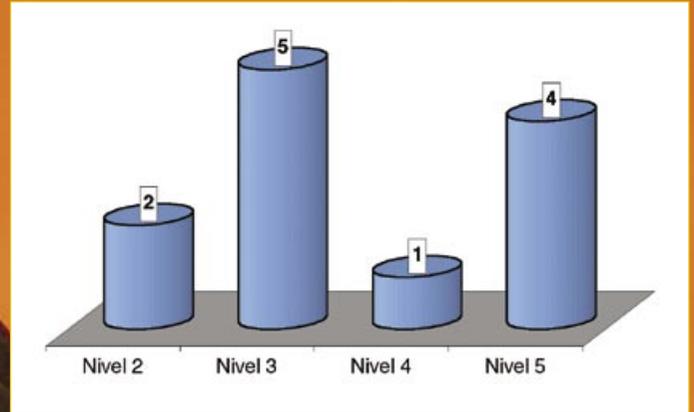
Como sabemos, existen diferentes opciones de modelos de referencia. La gráfica 2 ilustra la distribución de modelos elegidos.



Gráfica 2. Modelo de referencia utilizado.

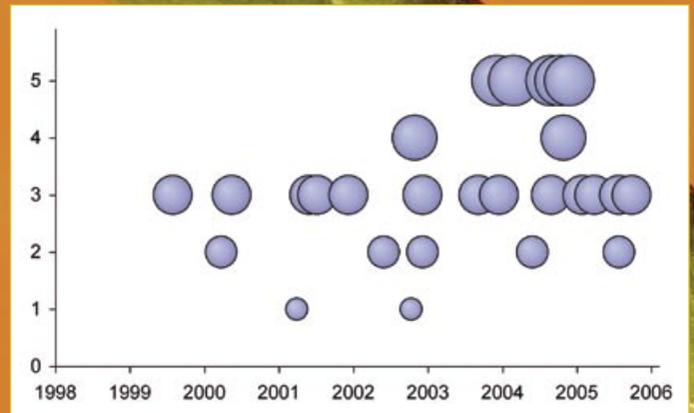
Evaluaciones

En México tenemos información sobre 12 organizaciones actualmente en operación, que han alcanzado diferentes niveles de madurez en evaluaciones realizadas de acuerdo con la información que se muestra en la gráfica 3.



Gráfica 3. Cantidad de empresas por nivel de madurez.

El total de evaluaciones CBA IPI (CMM based Appraisal for Internal Process Improvement) realizadas, es de 30 durante los últimos 6 años, con un mayor crecimiento en los últimos, según se muestra en la gráfica 4. Una de cada tres de estas evaluaciones es liderada por consultores nacionales. Adicionalmente a las evoluciones CBA IPI reportadas, se reporta una evaluación SCAMPI (Standard CMMI Appraisal Method for Process Improvement) de nivel 5 para el modelo CMMI en IBM.



Gráfica 4. Historial de evaluaciones CBA IPI.

Necesidades y Obstáculos

Las necesidades principales de estas organizaciones se orientan hacia la mejora de la calidad de los productos actuales, mejorar el control y administración de los proyectos, así como incrementar la satisfacción de los clientes. Otros intereses están enfocados hacia la integración de las iniciativas de mejora en la organización, mayor participación en el mercado de exportación, administración del conocimiento, y crecimiento en número de recursos.

Los principales obstáculos se presentan por la resistencia al cambio, la carencia de recursos y presupuesto para el proyecto y el desconocimiento de los modelos de referencia. También se presentan dificultades por la ausencia de cultura de procesos tanto en los clientes como usuarios, la falta de comunicación, pobre compromiso de la dirección, complejidad de los procesos y falta de capacitación en el uso de procesos y herramientas.

Carlos Pérez es socio fundador y director de consultoría en Avantare, donde ha colaborado con diversas empresas para lograr sus objetivos en el programa de mejora. Actualmente trabaja en proyectos con organizaciones orientadas a implantar niveles de madurez 3 y 4 de CMMI. Es expositor frecuente en seminarios y conferencias, instructor de cursos de ingeniería de software, así como del curso oficial "Introduction to CMMI" del SEI. Carlos es egresado de Ingeniería en Sistemas del ISPJAE en Cuba, y cuenta con una Maestría en Ciencias de la Computación de la UNAM.

Los beneficios se orientan a la mejora en la administración de proyectos, al establecimiento de la cultura de procesos, un mejor conocimiento del negocio, así como prácticas bien establecidas independientes de las personas que las ejecutan. Otros beneficios están relacionados con la participación en nuevos mercados, crecimiento sostenido, mejora en la satisfacción de los clientes, la calidad de los productos, y el reconocimiento por parte del cliente y el mercado por los logros alcanzados.

Toda esta información es de gran interés tanto para el mercado nacional como internacional, así como un mecanismo de promoción para las organizaciones que han

alcanzado exitosamente las metas en sus programas de mejora de procesos. Estudios como este pueden ser un punto de partida para presentar los resultados obtenidos en cuanto a la mejora de procesos en México y contribuyen al reconocimiento de la aceptación que han tenido en los últimos años los modelos de procesos, principalmente por el apoyo de los programas de gobierno. La publicación y continuidad del estudio se podría lograr a través de organismos independientes con mayor acceso a las fuentes de información.

Agradecemos a las empresas participantes por la información proporcionada y su contribución al estudio.

Escenario Internacional

De acuerdo con el último reporte publicado por el SEI con estadísticas sobre las evaluaciones realizadas basadas en el modelo SW CMM, los países con mayor número de evaluaciones y que además han alcanzado el nivel 5 de madurez son Estados Unidos, India y China, en ese orden de un total de 61 países que han reportado resultados. En ese reporte, México ocupa la posición número 13, encabezando la lista de países latinoamericanos por encima de Brasil y Chile. Por otra parte, en el último reporte de las evaluaciones basadas en el modelo CMMI, los países con mayor número de evaluaciones SCAMPI son Estados Unidos, India y Japón, igualmente con nivel 5 de madurez de un total de 41 países que han reportado resultados. En el caso de los países latinoamericanos Brasil encabeza la lista.

País	SW CMM		CMMI	
	Evaluaciones	Nivel máximo reportado	Evaluaciones	Nivel máximo reportado
Estados Unidos	1982	5	365	5
India	399	5	104	5
China	284	5	62	5
Japón	161	5	100	5
Francia	146	5	26	5
Reino Unido	140	3	29	4
Canadá	81	5		
Corea	77	5	30	5
Alemania	70	3	16	5
Australia	36	5	14	5
Italia	36	5		
Israel	31	3		
México	30	5	< 10	ND
Brasil	29	5	10	ND
Chile	20	5	< 10	ND
Argentina	< 10	ND	< 10	ND
Colombia	< 10	ND	< 10	ND
Costa Rica	< 10	ND		
Perú	< 10	ND		
Taiwan	< 10	ND	18	3
Uruguay	< 10	ND		
Venezuela	< 10	ND		

Tabla 1. Perfil de madurez de procesos por país. ND: No está disponible esa información. Sólo se publica para países que tienen más de 10 evaluaciones reportadas.

Esta información puede consultarse en la siguiente tabla resumida del "Process Maturity Profile" que publica el Software Engineering Institute (www.sei.cmu.edu).



En los últimos años he tenido la oportunidad de participar en aproximadamente 30 evaluaciones de procesos de software con empresas de distintos países de Latinoamérica. En estas evaluaciones, así como en mi experiencia como consultor de mejora de procesos, me ha llamado especialmente la atención el cambio que se opera en una empresa al pasar del Nivel 1 al Nivel 2.

En las empresas de Nivel 1, muchas veces sus ejecutivos no están concientes de sus males, asumiendo sus problemas como algo inmutable. Los administradores de estas organizaciones manejan el caos existente a través de un conjunto de prácticas típicas de estas empresas. Entre ellas están el dar continuamente explicaciones a sus clientes, y a su vez recibir de la misma forma explicaciones de sus jefes de proyectos, reducir las funcionalidades comprometidas para lidiar con los retrasos, bajar la calidad de sus productos hasta límites increíbles eliminando actividades de testing, despedir a los “culpables” de los proyectos fracasados, tener a toda la empresa en línea (léase: disponible en todo momento, especialmente en vacaciones), ser una empresa de “7x24 horas” donde la gente trabaja hasta altas horas de la noche e incluso fines de semana. Sin embargo, el peor síntoma es cuando los administradores no sólo asumen esta situación como algo natural, sino que están convencidos de que lo están haciendo estupendamente bien. En estas organizaciones, los gerentes se han convertido en expertos en el manejo de cada una de las contingencias anteriores. El problema es que han estado por años en lo mismo, han visto tantas empresas iguales, que poder reconocer el problema y la necesidad de cambio, les resulta extremadamente difícil y en muchos casos imposible.

Por lo anterior, el desafío más importante en el mejoramiento del proceso productivo de software está en pasar del Nivel 1 al 2. Ello requiere de una gran cuota de liderazgo de parte de la dirección. El Nivel 2 permite enfrentar las causas de fondo de los problemas, centrándose principalmente en los aspectos de la administración de los proyectos. La instauración de una disciplina de trabajo, a través de prácticas establecidas en el modelo CMMI, da a la organización la visibilidad necesaria que permite crear una infraestructura metodológica repetible, medible, auditable y mejorable en forma continúa.

CMMI Nivel 2

El Gran Desafío

Por José Guerrero C.

José Guerrero Cáceres es Socio-Gerente de América XXI, primera empresa chilena dedicada al mejoramiento de la calidad de software. Ha conducido más de 30 evaluaciones en CMM y participado en evaluaciones formales desde el Nivel 2 al Nivel 5 en Chile, Perú, Argentina y México. José es Ingeniero de Ejecución en Procesos de Información de la Universidad de Chile y es SEI Authorized Lead Asesor (# 0300924/A). jguerrero@americaxi.cl

Las organizaciones que alcanzan el Nivel 2 y superior, tienen una serie de características comunes entre sí. De hecho, creo que sería posible estimar el nivel de madurez de una organización en forma muy aproximada sin necesidad de revisar sus procesos. Bastaría con analizar en qué grado se dan las siguientes características en la organización que se desea evaluar:

Percepción del personal respecto de la administración. A medida que va aumentando el nivel de madurez de la organización, la percepción del personal en relación a aquellos que dirigen es cada vez mejor. Esto no es un aspecto menor. Permite a la organización ser un gran equipo de trabajo con una alta motivación y capacidad de logro.

Síndrome de no volver atrás. Una vez que se ha institucionalizado un proceso y la gente empieza a ver los beneficios, nadie añora volver al caos que existía antes. Nadie desea tener la “libertad” de hacer las cosas a su manera, de estimar sin métodos, de probar de acuerdo al tiempo disponible, etc. Todos saben el alto costo que se paga por estas prácticas. Un proverbio común en las empresas maduras, dice que “Quien haya trabajado en una empresa Nivel 2, jamás querrá volver a trabajar en una empresa Nivel 1”.

Calidad de vida. Una empresa que dispone de un proceso institucionalizado, mejora la calidad de vida de su personal. Las actividades y plazos son planificados de acuerdo a la disponibilidad real de los recursos.

Prestigio en el mercado. Las empresas CMMI gozan de un alto prestigio entre sus pares y en el mercado. No sólo por poseer una certificación reconocida en el mundo entero, sino porque lograr satisfacer las exigencias de CMMI requiere de una empresa bien dirigida, con empleados preparados, con metas claras, orientada al largo plazo, con sólidos valores y una alta autoexigencia.

Autovaloración del personal. El personal siente un legítimo orgullo de su trabajo y la forma en como lo realiza. Comprende lo que significa y se siente bien de trabajar en la empresa. Disfruta del trabajo que realiza y de los productos que entrega a los clientes. Tiene un muy buen ambiente de trabajo con sus pares y con la administración.

Resultados de la empresa. Los resultados de las empresas CMMI mejoran. Por una parte, pueden controlar mejor sus costos, plazos y recursos. Aumenta la productividad y la calidad de sus servicios y productos. Por otra parte, están en condiciones de competir por mejores proyectos, participan ventajosamente en las propuestas debido a la acreditación que poseen, se incorporan en forma creciente como proveedores de los clientes más relevantes de sus mercados objetivos.

Muchas organizaciones que están en el Nivel 1 tienen reticencia a incorporar procesos formales. Esta resistencia es producto de años de realizar las cosas de una determinada forma. Ello hace muy difícil poder imaginar que las cosas se pueden hacer de manera distinta y que funciona mucho mejor. El decidirse a probar una nueva forma aparece como un gran riesgo, que muchos no están dispuestos, o no pueden asumir. Sin embargo, aquellos que toman el camino del mejoramiento de procesos comprueban a poco andar que el riesgo no era tal y que los resultados pagan con largueza los esfuerzos realizados.

Entre los aspectos que me parecen más destacables en las empresas que dan el gran salto al Nivel 2, está el comprobar que han derribado una serie de mitos, que acostumbran estar presentes en las empresas Nivel 1.

Mito N°1: Esta es la mejor manera de hacer las cosas. Las organizaciones que dan el gran salto al Nivel 2, descubren que tras las montañas que parecían infranqueables hay un mundo mucho mejor que en el que vivieron tantos años. Un mundo lleno de oportunidades de crecimiento, un clima en el cual los empleados se sienten orgullosos de trabajar, métodos modernos de trabajo de los que todos se sienten parte, y el convencimiento de que está en cada uno de ellos seguir mejorando.

Mito N°2: Si agregamos todas estas nuevas actividades seremos más improductivos. Uno de los aspectos más reconfortantes de comprobar en las evaluaciones de Nivel 2 es que invariablemente todos los integrantes de la empresa estiman que, aún cuando ahora realizan una serie de actividades nuevas, su productividad no ha bajado y, por el contrario, ha aumentado. Ello es producto de hacer las cosas bien. Se elimina gran parte de la am-

bigüedad en los requerimientos, se controlan las piezas de software, se descubren los defectos antes, hay estimaciones realistas y, en definitiva, mucho menos retrabajo.

Mito N°3: Si ponemos plazos y costos más realistas, los clientes no nos comprarán. Parte de este mito ya se derribó en el punto anterior. Pero, aún cuando aparentemente hay un aumento de plazos y costos, las empresas comprueban que están en mucho mejor pie para negociarlos con el cliente. Contar con una sólida base de estimación y un proceso claramente definido, permite explicarle al cliente porqué se le está cobrando ese precio y dando ese plazo. Una vez que un cliente ha tenido la oportunidad de conocer su proceso, le es muy difícil asignar el proyecto a alguien que le ofrezca un precio y plazo menor, pero que es incapaz de explicar con claridad como lo va a lograr.

Mito N°4: No existe la motivación necesaria en el personal. Uno de los síntomas que busco cuando realizó una evaluación de Nivel 2, es saber cuál es la disposición de la gente en relación a la instauración de un proceso de desarrollo de software. En todos los casos, las personas demuestran gran aprecio por los resultados logrados con los nuevos procesos y declaran su decidido apoyo para colaborar en la implantación de otros procedimientos que les permitan alcanzar mayores niveles de madurez.

Mito N°5 Si hemos funcionado así, podemos seguir haciéndolo igual. Al establecer métodos formales, las organizaciones comprueban que ello les permite dar visibilidad y administrar sus proyectos, gestionar sus problemas, tomar acciones correctivas oportunas y, en muchos casos, evitar que los problemas ocurran. La forma tradicional, tarde o temprano, nos conduciría a un abismo.

Hoy en día, un gerente no puede soslayar el proceso de software que su organización utiliza para llevar adelante su tarea. Esto es parte integral del desarrollo de software. Hacerlo no es una opción, es una obligación. Las empresas CMMI son diferentes. Son altamente competitivas, pues poseen un grupo humano cohesionado entrenado, un sólido proceso para realizar su trabajo y están en permanente mejora de sus prácticas.

SPIN en América del Sur

Situación y Desafíos

Por Alejandro Bedini, Karin Steembecker y Miroslav Pavlovic



Han pasado veinte años desde que el término SPI (*Software Process Improvement*), o mejoramiento de procesos de software, se acuñara en el lenguaje de ingeniería de software. Cada día son más organizaciones dedicadas al desarrollo de software que aplican modelos de madurez, de calidad y/o metodologías para mejorar los procesos orientados al software, con la esperanza de mejorar la calidad de sus productos y aumentar la satisfacción de sus clientes.

A comienzos de 1990, nace la iniciativa en el SEI de crear centros independientes que permitan difundir el mejoramiento de los procesos de software a las empresas y organizaciones cercanas a ellos. A estos organismos se les conoce como SPIN (*Software Process Improvement Network*) o red de mejoramientos de procesos de software. Los SPIN son organizaciones sin fines de lucro que dependen exclusivamente de recursos externos para poder subsistir. Esta iniciativa tomó gran fuerza en USA, con la aparición de múltiples SPIN en diferentes estados. Con la expansión del modelo de madurez CMM a otros países, los centros SPIN se consolidan, crecen y fortalecen.

Pero, ¿qué pasa en nuestra región?, ¿han cumplido sus objetivos los SPIN regionales?, ¿han apoyado a las empresas locales para crear un círculo virtuoso de SPIN? En este artículo abordamos estas interrogantes.

SPIN en América del Sur

SPIN es una red de individuos y/o entidades que incentivan mejorar las prácticas en el desarrollo de software. Esta red promueve el uso de la ingeniería de software como herramienta base para aumentar la madurez de la empresa.

Actualmente existe una cantidad limitada de centros SPIN en la región, la tabla 1 indica los SPIN activos y emergentes en América del Sur, la cantidad en cada país y su nivel de actividad. El nivel de actividad presente en dichos centros fue establecido mediante entrevistas directas a presidentes y miembros, numerando la cantidad de actividades en desarrollo, y visitando su sitio web. Además de la contribución de uno de los autores de este artículo, que es fundador y co-fundador de 5 SPIN en la región.

País	SPINs Activos	SPINs Emergentes	Actividad
Argentina	0	1	Media
Bolivia	1	1	Baja
Brasil	11	3	Alta
Chile	1	0	Media
Colombia	1	1	Baja
Ecuador	2	0	Baja
Perú	0	1	Baja

Tabla 1. SPINs en América del Sur^[1].

Alejandro Bedini G. es consultor en Practia Consulting para la implantación de CMMI y SPICE. En su carrera ha asesorado empresas en diversos países de América y Europa. Es fundador y Presidente de SPIN – Latinoamericano, así como cofundador y colaborador de SPINs en Perú, Bolivia, Argentina y Colombia. Adicionalmente, es autor del libro "Gestión de proyectos de Software", orientado a las empresas de Latinoamérica. Alejandro es Ing. y Master en Ingeniería Informática en la Universidad Técnica Federico Santa María, Chile. spin@latinspin.org

SPIN como Herramienta Difusora de Iniciativas

Pero, ¿por qué es necesario un SPIN y qué beneficios conlleva? El SPIN es necesario como elemento motivador, fuente de conocimiento y negociador ante el gobierno para conseguir fondos o subsidios para apoyar las certificaciones.

Uno de los principios fundamentales para la evolución y vigencia de los SPINs es compartir experiencia, casos y apoyo mutuo entre estos organismos. Por tal motivo, se ha creado SPIN Latinoamérica (www.latinspin.org) para fortalecer los SPINs de la región. Una de sus primeras metas es traducir los modelos CMMI e ISO/IEC 15504, esperando que eliminar la barrera del idioma sirva de motivación para que las empresas mejoren sus procesos bajo estos modelos. SPIN cuenta con el patrocinio principal del SEI, ISO SPICE y actualmente se encuentra en negociaciones con el ESI.

Para las PyMEs desarrolladoras de software en América del Sur, la decisión de implantar un marco de trabajo orientado a la mejora de procesos no es algo trivial, a pesar de los beneficios concretos que se han demostrado. La principal variable de decisión es la inversión inicial necesaria para llevar a cabo un plan de mejora. La respuesta que facilitaría esta decisión es la "**Coopetencia**". Este concepto surge del e-business, y se refiere a la cooperación entre empresas para lograr un objetivo en común con beneficios mutuos pero con competencia hacia el cliente final [2]. De esta forma se combinan las reglas de la competencia y la cooperación en una nueva dinámica que puede ser usada no sólo para incrementar las utilidades, sino también para mejorar la naturaleza del ambiente de los negocios. Por consiguiente, lo mejor que pueden hacer las empresas en América del Sur es unirse bajo un objetivo común que es mejorar sus procesos de software. **SPIN permite realizar una coopetencia entre empresas y gobierno para conseguir fondos en programas de mejora.**

Entonces el dicho la "unión hace la fuerza", se podría traducir en "**la unión prorroga costos y permite gestionar apoyo del gobierno**".

Lecciones Aprendidas

Las observaciones encontradas durante este ejercicio y complementarias a estudios del ESI en la región, que guardan relación con los errores más comúnmente cometidos al momento de implantar un SPIN son las siguientes^[3]:

Errores

- Crear un SPIN y esperar a que las personas lleguen solas.
- No hacer un estudio de factibilidad económica para mantener SPIN.
- No realizar difusión de su existencia.
- No establecer metas.
- Pedir una cuota de incorporación aclarar qué se recibe a cambio.

Mejores Prácticas

- Conseguir patrocinador, pueden ser empresas y/o universidades.
- Incorporar a estudiantes como soporte al SPIN. Serán los futuros ingenieros de calidad que apoyaran al SPIN.
- Difusión en los medios.
- Actividades programadas de charlas, talleres y cursos.
- Buena relación entre la cuota cobrada y servicios recibidos.
- Agente catalizador de SPI y acuerdos gobierno-empresa.
- Desarrollo de estudio de mercado sobre SPI para las empresas.

Conclusiones

La continua e ineludible comparación entre países de similar actividad y desarrollo tecnológico, exacerbada por el bajo costo en la mano de obra que ofrecen nuestros países a las grandes economías del mundo, hace imperativa la necesidad de mejorar los actuales procesos de desarrollo de software observados en la región. Afortunadamente, la iniciativa SPIN ha tomado cartas en el asunto, instalándose en siete países y creciendo. Sin embargo, el verdadero impulso que requieren estos organismos para establecerse en la región y convertirse en una alternativa viable son:

- a) Ver la exportación de software como una actividad prioritaria y de generación de valor superior por parte del gobierno.

b) Fortalecer los centros de SPIN como entidad creadora de espacios de comunicación entre gobierno-empresa-universidades-profesionales.

c) Compartir experiencias entre los países y organizaciones locales e incentivar que las empresas se unan para encarar conjuntamente proyectos de este tipo. Este es el objetivo de Latin SPIN.

De este estudio se desprende que la situación general de los SPIN en la región es de Crisis. Son muy pocos los casos de éxito rotundo en este tipo de organizaciones, donde la mayoría están en un estado más de supervivencia que de sinergia. Las expectativas podrían ser promisorias por el aumento en países como Brasil, Chile, Argentina y Perú (con 90 empresas en proceso de implantar SPI) en vías de evaluación, sobre todo, del modelo CMMI. Debe tomarse esta situación como una perspectiva de mercado que aliena los SPIN.

Para realizar este artículo se contó con el universo de estudio que contiene una muestra de 35 documentos oficiales e informes, referidos al lanzamiento de los respectivos SPINs, actividades logradas, cantidad de socios, y publicaciones relacionadas al SPI en la región. Además, se contactaron vía e-mail o personalmente a más de 23 miembros de varios SPIN de la región. Adicionalmente, se consideran las apreciaciones personales de los autores. Es importante mencionar que uno de los autores es fundador de varios SPIN en la región.

Referencias

1. Software Process Improvement Networks www.sei.cmu.edu/collaborating/spins/spins.html
2. Alejandro Bedini. Congreso Internacional de Gestión del Conocimiento y la Calidad. Cartagena de Indias, Colombia. 2004.
3. Jose Árias, Presentación en SPIN Chile. Santiago, Chile. 2004.

Karin Steembecker es consultora en Practia Consulting, donde ha llevado a diversas empresas a nivel 3 del modelo CMMI. Karin es Ing. en Informática de la Universidad de Santiago de Chile, y posee un master de la Leeds Metropolitan University en Inglaterra. Además es profesora invitada del master en informática de la USACH en Chile. ksteebecker@pragmaconsultores.com

Miroslav Pavlovic es consultor en Practia Consulting, trabajando con metodologías de desarrollo, modelos de calidad y certificaciones basadas en CMM y CMMI. Ha participado en el desarrollo del estándar "IEEE P1074 (12207) - Standard for Software Development Process Lifecycle", y es miembro del "Comité de Revisión de Estándares de Ingeniería de Software" de la IEEE. Miroslav es graduado y postgrado en la Universidad de Belgrado. mpavlovic@pragmaconsultores.com

La Calidad no Basta Innovar para ser Competente

Por Francisco Rivera Martínez, PMP

➤➤ Actualmente se está haciendo un esfuerzo —apoyado con fondos locales y federales— para formar y desarrollar clusters de Software en un contexto de mejora de la economía del país. Existen ya agrupaciones de empresas de este tipo en varias entidades como Guanajuato, Jalisco, La Laguna, Nuevo León, Distrito Federal, etc.

La estrategia seguida en ellos es prioritariamente en el sentido de incrementar la madurez de los procesos, basándose en modelos como CMM y MoProSoft. Este esfuerzo, de ser exitoso, llevará en unos años de trabajo e inversión a las empresas participantes al nivel de “confiable” (ver tabla 1) o repetible, lo que les permitirá competir de esa manera al mejorar su nivel profesional. Sin embargo, esto no necesariamente las llevará al nivel “competente”, que es lo que necesitan para salir de los segmentos de mercado donde se compite por precio, y acceder a segmentos de mayor valor y con mayores barreras de entrada para sus competidores.

Características en la empresa	Nivel de competitividad			
	Emergente	Confiable	Competente	Clase mundial
PRIORIDAD	Supervivencia	Diferenciación	Innovación	Liderazgo
MEJORES PRÁCTICAS	Sistemas gerenciales y administrativos	Mejora continua y benchmarking	Desarrollo de nuevos productos	Obsolescencia acelerada de productos
COBERTURA DE MERCADO	Local	Nacional	Región internacional	Global
NIVEL DISTINTIVO DE ADMINISTRACIÓN	Operación	Calidad o exportación	Calidad y exportación	Gestión tecnológica
CAPACIDAD TECNOLÓGICA	Imitación	Adopción y/o mejora	Desarrollo	Licenciamiento
ACTITUD AL CAMBIO	Reacciona	Se adapta	Promueve	Origina

Tabla 1. Nivel de competitividad de las empresas.

Innovación

La innovación tecnológica, prioridad del nivel “competente”, consiste en llevar una tecnología o conocimiento al mercado que nos interesa para aportar un mayor valor al cliente o usuario de nuestros productos y servicios. Por ejemplo, si una empresa está desarrollando un sistema de reservaciones para un parque de diversiones, no sólo debe lograr que el producto efectivamente permita a los usuarios reservar fácilmente y al encargado del parque administrar estas reservaciones, sino diseñar los paquetes de diversiones a reservar de manera tal que se haga un óptimo uso de los recursos disponibles, maximizando su ocupación y la utilidad, mediante técnicas de investigación de operaciones. De esa manera, está añadiendo mayor valor, su producto será más apreciado, se venderá a mejor precio y no cualquier alumno de tercer semestre podrá ganarle el cliente. Si adicionalmente la tecnología que incorpora es una tecnología que tiene esa empresa en exclusiva y la tiene protegida, sólo ella podrá ofrecer ese valor agregado.

El Dr. Francisco Rivera Martínez es consultor empresarial para la innovación tecnológica en el Instituto de Tecnología de Software, S.C., profesor en el ITESO, y colabora en Innevo como instructor. Es Ing. Mecánico Electricista del Iteso y cuenta con un doctorado de la Université Paul Sabatier de Toulouse, Francia. Francisco es Project Management Professional (PMP) certificado por el PMI y evaluador acreditado en el RCEA del CONACyT en las áreas de administración y sistemas, control y aseguramiento de calidad.

Desarrollo de Producto

El desarrollo de nuevos productos con un componente de innovación tecnológica, básicamente trata de un peloteo de ideas e información entre los aspectos de mercadotecnia y de tecnología relativos al producto de que se trate —según lo proponen los métodos conocidos de innovación, como el de Myers y otros—, donde se trata de tener, por un lado, un mejor conocimiento de las características y necesidades del segmento de mercado al que se pretende llegar y, por el otro, de las tecnologías que permitirán satisfacer mejor las necesidades detectadas. En consecuencia, para hacer nuevos productos innovadores, hace falta —entre otros elementos— disponer de procesos de búsqueda de información tanto de mercado como tecnológica.

Las búsquedas de información de mercado pueden incluir estudios de conocimiento de clientes y mercados, de competitividad de productos y servicios propios y de la competencia, etc.; las búsquedas tecnológicas pueden ser de artículos científicos y tecnológicos, análisis de patentes, reportes técnicos, manuales de proveedores, etc. Adicionalmente, conviene estar informado de lo que está surgiendo en ciencia y tecnología en el campo que se trabaja, para prever el futuro, haciendo lo que se ha dado en llamar vigilancia tecnológica.

El punto está en que los clusters de software como los que hay en México, están formados principalmente por empresas pequeñas que difícilmente tendrán áreas de mercadotecnia e investigación, lo que les dificulta hacer desarrollo de producto; sin embargo, justamente al formar agrupaciones, éstas les deberían permitir crear estos procesos —y otros— de manera centralizada, para que todos los miembros del cluster reciban sus servicios a un costo que puedan asumir.

Planeación Estratégica

Las empresas —y agrupaciones— deben contar con una planeación estratégica y tecnológica, que les permita saber, entre otras cosas, con qué recursos cuentan, qué oportunidades se les presentan, y qué deben hacer para aprovecharlas. De esta forma podrán seleccionar adecuadamente los procesos clave en que deben desarrollar mejores prácticas. Mi percepción es que muchas empresas actualmente están adoptando modelos de calidad de software de una manera subjetiva, y no claramente intencionada. Como una sugerencia, y de manera complementaria a los procesos de software que contemplan los modelos de calidad preestablecidos, pudiera analizarse la pertinencia de incluir en sus programas de capacitación

y consultoría los procesos asociados al desarrollo de productos innovadores, como son:

- Diagnóstico tecnológico.
- Planificación estratégica y tecnológica.
- Vigilancia tecnológica e inteligencia competitiva.
- Búsqueda y selección de alternativas tecnológicas.
- Redacción y negociación de contratos.
- Gestión de la propiedad intelectual.
- Avalúo de tecnología.
- Gestión de fondos.

Esfuerzo Doble

Es un aspecto polémico, pero se debería intentar pasar del estado de empresa “emergente” —en el cual se hallan una parte significativa de los miembros de los clusters mexicanos—, a un estado donde se combinen las prácticas propias de los estados de empresa “confiable” y “competente”. Se podría hacer un esfuerzo simultáneo en ambos sentidos: mejora en la madurez de procesos por medio de la implantación de algunas de las mejores prácticas incluidas en los modelos nacionales e internacionales; y mejora en la competitividad de sus productos y servicios por medio de la innovación tecnológica, para no perder más tiempo y lograr pasar a una mayor competitividad empresarial en un tiempo razonable.

Cluster Innovador

Una respuesta a esta situación podría ser el concepto de Cluster de Software Innovador, donde además de las empresas se podría invitar a participar a universidades, institutos de educación superior, centros de investigación, empresas de consultoría tecnológica y de negocios, despachos de abogados, gestores de proyectos tecnológicos, etc. y a todos los actores de lo que se conoce como el ecosistema tecnológico de una región determinada. De manera tal que se forme una masa crítica de participantes con el fin de mejorar la competitividad de la planta de empresas de TIs, lo que redundará en beneficio para todos.

Transferencia Tecnológica

En el país existen proveedores de tecnología confiables, como es el caso de investigadores en la UNAM, que podrían ser buenos agentes de cambio de los clusters mediante transferencias tecnológicas en el área de gestión de la innovación y la tecnología, para con su apoyo acelerar el proceso hacia una mayor competitividad. ©

Ingeniería Mexicana
de Software S.A. de C.V.



*Desarrollando
para una producción
más EFICIENTE*

Somos una empresa especializada en el desarrollo de software a la medida, nuestra metodología se basa en la aplicación de las mejores prácticas para el desarrollo y el uso de herramientas CASE para desarrollo y auditoría del software.

También contamos con modelos o productos base, los cuales son fácilmente adaptables a las necesidades específicas de su empresa, algunas de esas soluciones incluyen aplicaciones para Facturación, Punto de Venta, Almacenes, CRM, Control de Obras, Activos Fijos, Biblioteca, Videoteca, etc.

Contamos con especialistas para ofrecer soluciones con distintos manejadores de bases de datos, lenguajes de programación y plataformas desde AS 400 hasta Pocket PC.

www.imexsoft.com.mx
webmaster@imexsoft.com.mx

**Torres Adalid #707
despacho 802
Col. Del Valle, México, D.F.
C.P. 03100
Tel:(55) 5687 5052**

Modelos de Procesos para la PyME

¿CONOCEN LAS EMPRESAS LATINOAMERICANAS ESTOS MODELOS?



Irma María García Barranco es profesora de planta en el Departamento de Sistemas de Información del Tec de Monterrey, Campus Estado de México. Fue coordinadora del proyecto de automatización del Centro de Atención a Alumnos en esta institución. Irma es Ing. en Sistemas Computacionales y cuenta con una Maestría en Administración de Tecnologías de Información.

En los últimos años se ha dado una mayor competencia entre las empresas de desarrollo de software. En el contexto de Latinoamérica, en su mayoría son pequeñas empresas, que aparecen y desaparecen con relativa facilidad. De primera instancia se podría pensar que se debe a que no hay suficiente demanda para este sector. Sin embargo, esto dista de ser cierto. El mercado interno tiene una oferta importante para la pequeña empresa productora de software. El verdadero problema es que las empresas no tienen procesos definidos.

No se trata de inventar algo nuevo, en todos los sectores se ha buscado la orientación a procesos, debido a que es más fácil la evaluación y posteriormente emprender acciones de mejora, ya que a través de cada proceso están bien definidos los insumos necesarios para que se lleven a cabo las actividades, así como los resultados que se esperan al finalizar. Estos resultados, pueden ser medibles y evaluables.

Existen diversos modelos de procesos de software, que definen procesos clave y actividades a realizar dentro de las organizaciones. Entre los más conocidos están PSP, TSP, CMMI, IEEE 12207. La pregunta importante es, ¿conocen las empresas estas opciones?

En general, las pequeñas empresas de software están poco familiarizadas con estándares de procesos. Aunque no son modelos nuevos —algunos tienen cerca de 20 años—, es hasta fechas recientes que ha surgido la preocupación de conocerlos e implementarlos. Los organismos de gobierno e instituciones educativas han jugado un papel importante para dar a conocer estos modelos y orientar a las empresas para que los consideren en sus prácticas. Sin embargo, el problema que surge de ello es la falta de infraestructura para soportar estas prácticas o esta organización.

Considerando el estándar IEEE 12207, se propone una estructura de 5 procesos primarios (enfocados a la razón de ser de una empresa de software) y procesos de soporte enfocados a la calidad del proceso, y para garantizar la calidad de producto; tiene la ventaja de ser un estándar muy general, ya que con ello las empresas lo pueden adaptar de acuerdo a la estructura organizacional y a los recursos con los que cuentan. Sin embargo, las empresas latinoamericanas de desarrollo de software tienen menos de 10 empleados en su gran mayoría, esto nos lleva a determinar si realmente se pueden establecer todos los procesos, porque la carga administrativa se vería realmente incrementada.

Realmente es muy costoso para las organizaciones de Latinoamérica construir una organización tan compleja, orientada a procesos tales como la adquisición, suministro,

desarrollo, operación y mantenimiento; además de procesos de soporte como documentación, administración de la configuración, aseguramiento de calidad y solución de problemas. La realidad es que las empresas de software únicamente desarrollan, y hacen un poco o nada de administración de proyectos; y realizan estas actividades sin planeación, sin control, y pocas veces están estandarizadas.

Claro está que IEEE ofrece esto como marco de referencia, pero la orientación a procesos dentro de las organizaciones de desarrollo de software ha representado un reto internacional, implica cambios culturales, infraestructura, capacitación y, como ya se ha mencionado, las empresas latinoamericanas no cuentan con los recursos suficientes para impulsar estos cambios. Se han dado algunas iniciativas de gobierno con la finalidad de apoyar este tipo de industria y a través de la formación y apoyos económicos se logre que las empresas orienten sus actividades a procesos de acuerdo a sus capacidades y que estos sean optimizados a través de la mejora continua, para que se conviertan en empresas competitivas.

En México, sabemos que el programa ProSoft de la Secretaría de Economía está apoyando con fondos financieros a las PyMEs de este sector, para que se capaciten, cuenten con los recursos necesarios para realizar inversión en tecnología, y más que nada que exista un modelo de calidad enfocado a la realidad de las organizaciones de México, las cuáles no pueden considerar modelos tan complejos como los inicialmente mencionados. Se busca que estas empresas cumplan con estándares internacionales de calidad en procesos, para garantizar la calidad del producto, para poder competir con las empresas que cada día van surgiendo a nivel mundial, y estando conscientes de que es un sector que debe crecer porque está siendo necesario para las necesidades de otros sectores a nivel mundial. Es así que se ha forjado el estándar MoProSoft, Este estándar contempla la realidad de la pequeña y mediana empresa en el país, siendo la norma oficial que las empresas están adoptando. MoProSoft se ocupa entonces de esta preocupación, y sus promotores ahora enfilan sus esfuerzos a lograr su aceptación entre las empresas.

Nos encontramos entonces ante dos retos: uno es lograr que la pequeña empresa productora de software comprenda la utilidad de los modelos de calidad y esté dispuesta a adoptarlos. El segundo reto, aún más grande, es lograr la creación y adaptación de modelos, tales como MoProSoft, que sean factibles para las empresas y que les permitan la definición de procesos sin la carga administrativa de los modelos establecidos. Cumpliendo estos dos retos tendremos sin duda empresas de software más ágiles y productivas. ☺

- Irma García B.

Podrías tener mejores resultados...



Rational. software

Optimización de Procesos de Ingeniería de Software y TI

- Rational Unified Process
- CMMI
- CMMI by RUP
- ITIL
- Instalación y Configuración de herramientas Rational
- Optimización del área de Ingeniería de Software
- Alineación de objetivos de negocio con los servicios de TI
- Consultoría y capacitación en procesos de TI
- Sarbanes-Oxley
- Oficina de Proyectos

itera
it & development process

Adoptar un proceso de ingeniería de software confiable, eficiente y basado en las mejores prácticas de la industria • Asegurar el desarrollo exitoso y con calidad de proyectos de software • Mejorar la productividad de las operaciones de TI • Elevar el nivel de satisfacción de los servicios de TI • Facilitar la administración de la infraestructura de TI • Reducir el tiempo de implementación de procesos de mejora • Obtener visibilidad clara del desempeño de los procesos y proyectos de TI • Facilitar la toma de decisiones basada en información cuantitativa.

INSCRÍBETE

al Seminario Gratuito
"Mejorando los Servicios de TI para el Negocio y Automatizando la Administración de los Servicios de TI"

Cd. de México - 18 de Noviembre

Confirme su asistencia en:
contactsalescenter@itera.com.mx
(55) 5281 7670



Tivoli. software



www.itera.com.mx

Centro de Formación

- Rational University
- CMMI
- ITIL

ebXML

PARTE 2. FUNCIONAMIENTO EN UN SISTEMA COMPLETO

Por Atanacio Reyes

En el número anterior, dimos un contexto general sobre lo que es ebXML y sus principales componentes. En esta ocasión, mostramos con mayor detalle el funcionamiento de ebXML en un sistema, así como su relación con RosettaNet y Web Services.

La orientación de la arquitectura técnica de ebXML se enfoca en proveer una estructura para que las organizaciones puedan colaborar en la integración de sus procesos externos. Típicamente el primer paso de este proceso consiste en que una organización defina y publique sus procesos de negocio en un repositorio donde otras organizaciones puedan descubrirlos. Una vez que otra organización los descubre, basándose en ellos y en su propia definición de procesos, inicia una secuencia de negociación mediante la cual se establecen acuerdos de colaboración, y apegándose a los términos y condiciones de estos acuerdos, los socios de negocios crean y configuran sus interfaces electrónicas de servicios, con lo cual se habilitan para iniciar transacciones y ejecutar así los procesos acordados. Los procesos definidos en la primera fase de este escenario y que a su vez han sido acordados con otros socios de negocios deben administrarse y verificar que satisfacen los acuerdos. Conforme los participantes en la comunidad de negocios electrónicos aumentan, se evalúan los procesos existentes con el propósito de mejorarlos y se definen nuevos procesos con el fin de ajustarse a las necesidades del mercado. La secuencia descrita en el escenario anterior, necesaria para el establecimiento de una comunidad con procesos externos compartidos e integrados entre sí, puede perfectamente llevarse a cabo mediante la implementación de las especificaciones que forman la estructura ebXML. En este artículo se muestra la correspondencia entre los pasos del proceso descrito y cada una de las especificaciones ebXML.

Definición de Procesos de Negocio

Determinar qué procesos son necesarios para ser tomados en cuenta en una estrategia de comercio electrónico, además de la descripción de cada una de las actividades y elementos que los forman, es una tarea que puede ser realizada con mayor facilidad si se emplea una metodología orientada a capturar el conocimiento de una organización en prácticas de comercio

electrónico. UMM fue desarrollada con ese propósito, además de que ebXML la recomienda debido a que en las especificaciones BPSS y CPP/A existe una correspondencia con esta metodología.

Exhibición de las Capacidades, Servicios y Procesos

El modelado de procesos, necesario para describir y definir las capacidades de colaboración de un socio de negocios, es normalmente independiente de la tecnología. Sin embargo, para poder exhibir dichas capacidades en un repositorio público de tal forma que puedan ser encontradas por socios potenciales, es necesario exponerlas en un formato estándar entendible electrónicamente y capaz de ser procesado e interpretado por una máquina. ebXML define dos esquemas XML para este propósito: CPP y BPSS. El documento principal que describe las capacidades de un socio de negocios es el CPP, dichas capacidades están definidas en términos de roles que desarrolla, empaquetado de los mensajes, protocolos de transporte que es capaz de desarrollar, mecanismos de seguridad y procesos que ejecuta. La definición de los procesos se especifica en otro documento, en el CPP sólo se incluyen referencias a él. Aunque ebXML provee la especificación BPSS para la definición de procesos de negocios, no limita a que en el CPP se incluyan referencias a documentos formados con esquemas diferentes, que pueden ser XLANG, WSFL o PIP (Rosetta Net).

En cuanto a la definición de contenido de los mensajes que intervienen en las

colaboraciones, ebXML no especifica ninguna recomendación, dejando esta responsabilidad a otros organismos, los cuales pueden emitir formatos más apegados al giro de los negocios involucrados. Actualmente existen asociaciones que han definido contenido de mensajes para intercambio electrónico, como lo son: EDIFACT, ANSI, XEDI, Automotive Industry Action Group (AIAG), BOLERO, NET, OAG y otros.

Para almacenar el CPP y el documento que describe los procesos (que puede ser definido mediante el BPSS) en un repositorio, se utilizan los servicios de la interfaz Life Cycle Management (LM), provista por el registro. Para tener acceso a esta interfaz es necesario ser un usuario registrado en dicho repositorio. Un registro/repositorio normalmente es desarrollado por comunidades de empresas que se asocian con el fin de integrar sus procesos de negocios, sin embargo también existen empresas que proveen servicios de hospedaje en sus repositorios además de herramientas de software para transferencia electrónica de mensajes, modelado de procesos, transformación entre formatos XML etc.

Establecimiento de Acuerdos con Socios

Para encontrar socios de negocios potenciales, las empresas acceden a los servicios que ofrece el registro/repositorio a través de la interfaz Query Management. Para acceder a esta interfaz, no es necesario ser un usuario registrado del repositorio. Una vez que se descubre un posible socio en el re-

Atanacio Reyes Valenzuela es Lic. en Ciencias Computacionales de la Universidad Autónoma de B.C., tiene más de 10 años de experiencia en el desarrollo de software para integración de procesos de manufactura, interconectividad entre maquinaria y equipo para la automatización, procesamiento, control, administración, cálculo geométrico y matemático. Actualmente trabaja para la empresa Augen Ópticos, dedicada a la industria óptica oftálmica en México.

gistro, se puede transferir el CPP encontrado al servidor propio, con el cual se forma un CPA. Este CPA se envía al socio en cuestión para iniciar un proceso de negociación. Una vez que ambos socios están de acuerdo en el CPA, deben almacenar copias idénticas de este en sus servidores y con ellas configurar sus aplicaciones e interfaces con el fin de llevar a cabo los procesos acordados.

El CPA viene a ser un subconjunto de un CPP o puede ser la intersección entre los dos CPP's a partir de los cuales se construye (el CPP propio y el CPP obtenido del repositorio) y define la forma en la que dos socios de negocios interactúan en la realización de las colaboraciones que han acordado, la interacción también puede llevarse a cabo a través de intermediarios, para lo cual debe existir un CPA entre cada parte y el intermediario (adicionalmente al CPA entre ambas partes).

Al igual que el CPP, el CPA contiene referencias a la especificación de procesos (definida en BPSS, XLANG, WSFL o PIP de Rosetta Net), la que define una o más conversaciones entre las partes involucradas. La conversación representa una unidad simple de negocios definida por una colaboración binaria. Una conversación consiste de una o más transacciones de negocios, cada transacción es un mensaje de petición emitida por una de las partes que participan en la colaboración, y cero o un mensaje de respuesta emitido por la otra parte. Además la especificación de procesos define el orden en el cual las transacciones deben efectuarse (coreografía).

El CPA puede implementarse en un servidor B2B en el sitio de cada socio, donde deben residir sus especificaciones de procesos. Debe incluir además de los servicios que provee, una bitácora de auditoría. La información estática puede ser almacenada en una BD local, para generar los datos contenidos en las transacciones necesarias para satisfacer el CPA.

Intercambio de Información

Como se mencionó en el artículo anterior, ebXML especifica SWA como método de empaquetado para el transporte de mensajes, el propósito inicial de esta especificación, es que quienes estén usando EDI puedan realizar la transición a ebXML sin complicaciones. Como beneficio adicional permite que el contenido

útil del mensaje sea definido de forma libre y de acuerdo a las necesidades de las partes que intervienen en determinada colaboración. El cuerpo del mensaje SOAP contiene un listado de todos los documentos adjuntos al mensaje, el formato de cada documento puede ser XML, pdf, imágenes etc., siempre y cuando el formato se haya especificado en el CPA al cual el mensaje SOAP hace referencia. El destinatario del mensaje puede usar este listado para verificar si todos los documentos adjuntos se recibieron. Además, en el encabezado del mensaje SOAP se define un identificador de conversacion (ConversationId) para indicar la parte del proceso que se ejecuta una vez terminada exitosamente la transacción.

Administración de Procesos

Los pasos anteriores son los necesarios para completar un proceso de integración de procesos de negocios interempresarial. Como puede observarse, sólo dos especificaciones de la estructura de ebXML son necesarias para que dos partes puedan integrar sus aplicaciones en un esquema de colaboración: el CPA y ebMS. El CPA es el acuerdo que convierte a las partes en socios, cuyo fin común es la integración de sus aplicaciones. La especificación de mensajes (ebMS) es el medio a través del cual llevan a cabo sus transacciones.

Conforme la comunidad de negocios va creciendo, la dificultad para administrar los procesos de colaboración entre socios se incrementa. Suponiendo que se tienen 5 socios y con cada socio se ejecutan 3 procesos y cada proceso tiene 4 actividades, en total deben administrarse 60 actividades. Considerando que cada proceso puede tardar desde unas horas hasta varios días en completarse, monitorear el estado de cada proceso conlleva algunas complicaciones.

Considerando el problema de administración de procesos ejecutables, surge una tecnología para ayudar a las empresas a administrar sus procesos a fin de optimizarlos y adaptar el comportamiento de las empresas a los cambios que normalmente son necesarios. Estos sistemas se conocen como Business Process Management Systems (BPMS) —ver SG Año 1 No. 4—, y aunque ebXML no recomienda un modelo particular de implementación, este tipo de sistemas deben ser considerados al planear proyectos que involucran procesos ejecutables.

Lo que bien empieza, bien acaba.

+ de 60 proyectos exitosos de Calidad de Software en 2 años

- ⊖ Optimización de procesos de Desarrollo de Software
- ⊕ Acreditaciones de CMMI
- ⊖ Consultoría, cursos y talleres: CMMI y MoProSoft
- ⊕ Kanav

GDL DF MTY
(33) 31.20.29.99 (55) 50.93.46.54 (81) 83.49.76.14

www.innevo.com
info@innevo.com

Las dos especificaciones necesarias para que dos partes integren sus aplicaciones en un esquema de colaboración, son el CPA y ebMS

Relación de ebXML con RosettaNet

El consorcio RosettaNet (www.rosettanet.org) inició en Junio de 1998 auspiciado por la industria electrónica. El objetivo de la iniciativa es administrar de forma eficiente el canal de suministro a través del Internet. RosettaNet es un consorcio pionero en la definición de estándares basados en XML para integración B2B, y constituye la fuente de inspiración para ebXML. El modelo de esta especificación se divide en tres áreas:

1. Intercambio de mensajes. RosettaNet Implementation Framework (RNIF)
2. Definición de procesos. Partner Interface Processes (PIPs).
3. Conjuntos de datos usados en giros específicos de la industria, tales como códigos de productos, códigos de industrias, etc.

RNIF v2.0 es una especificación muy parecida a la de ebMS, lo cual le permitió al consorcio RosettaNet alinearse completamente con ebXML en la versión 3 de RNIF.

En cuanto a la definición de procesos, RosettaNet ha desarrollado un catálogo bastante amplio. Cada proceso (PIP) está formado por diálogos especializados sistema a sistema en XML. Cada especificación de PIP incluye un documento de negocios con su correspondiente vocabulario y un proceso con la coreografía de los mensajes. Los PIPs están agrupados en siete áreas:

1. Captura de información, mantenimiento y distribución para el desarrollo de CPP y suscripciones a información de productos.
2. Distribución y actualizaciones periódicas de especificaciones técnicas.
3. Precios, tiempos de entrega, recepción y administración de órdenes.
4. Administración de inventario.
5. Campañas de información y mercado-tecnia.
6. Asistencia técnica posterior a la venta.
7. Intercambios de diseño, configuración, procesos e información relacionada con manufactura.

Estas especificaciones de procesos son útiles para que las empresas que requieran integrar sus aplicaciones en un ambiente B2B escojan de entre este catálogo los procesos que mejor se ajusten a sus necesidades para definir sus CPPs. A partir del 2003 RosettaNet emplea la especificación de BPSS para definir todo su catálogo de procesos.

Relación de ebXML con Web Services

Existen muchas semejanzas que provocan confusión entre ebXML y Web Services. De hecho muchas empresas que ofrecen soluciones B2B e implementan en ellas especificaciones ebXML las anuncian como Web Services, colaborando con ello a aumentar la confusión.

Las principales semejanzas entre ambas tecnologías son:

- Usan el mecanismo de petición/respuesta sobre protocolos de transporte sin conexión permanente (HTTP, SMTP).
- Ambas usan SOAP como métodos de empaquetamiento de mensajes.
- El método para descubrir a las otras partes con quienes realizar transacciones es a través de un registro público (UDDI, registro/repositorio).
- ebXML especifica que la interfaz de un registro/repositorio pueda exponerse mediante servicios, lo cual provoca que las reglas de interacción con el registro se establezcan en un documento WSDL que puede ser descubierto mediante un UDDI.

Sin embargo, en los detalles de estas que parecieran ser semejanzas, existen profundas diferencias:

- El método de empaquetamiento de ebXML adiciona a SOAP extensiones que permiten la seguridad y confiabilidad de los mensajes usados en las transferencias, además de que es posible intercambiar documentos en cualquier formato convenido, no necesariamente XML.
- UDDI no provee almacenamiento de documentos, sólo provee referencias (links) a documentos WSDL, mientras que la separación entre registro y repositorio establecida por ebXML permite el almacenamiento de

cualquier tipo de documentos. Este esquema puede ser usado para que una organización respalde todo el conocimiento adquirido, sus metadatos, la descripción de sus procesos etc., en un repositorio interno.

- La unidad atómica de Web Services es un servicio (transacción), mientras que en ebXML es una colaboración (proceso o conversación). Una colaboración está formada por una o varias transacciones que se realizan siguiendo una secuencia preestablecida, por lo cual es posible definir colaboraciones partiendo de servicios. Para agrupar servicios dentro de colaboraciones y definir el flujo de los mismos, se han establecido las especificaciones XLANG y WSFL propuestas por Microsoft e IBM respectivamente. Aunque estas empresas están entre los principales promotores de la tecnología de Web Services, no han podido establecer una especificación común para el flujo de los servicios que componen las colaboraciones, por lo cual sus respectivos productos y soluciones de integración B2B resultan ser incompatibles.

Aunque en el esquema de Web Services sea posible establecer colaboraciones partiendo de servicios, el hecho de que la unidad atómica de esta tecnología sea el servicio, compromete mucho la confiabilidad de las colaboraciones, más aún tratándose de que el protocolo de transporte HTTP no es orientado a conexión. Además de que el tiempo necesario para que una colaboración sea ejecutada puede tomar varios días.

Conclusión

En estos artículos hemos visto a grandes rasgos de lo que se trata ebXML. Es una tecnología bastante prometedora, que posiblemente ayude a evitar que las transacciones de negocio por Internet se conviertan en una torre de Babel. Los invitamos a que conozcan más sobre ebXML y analicen si su adopción puede traer beneficios a su empresa o industria. ©

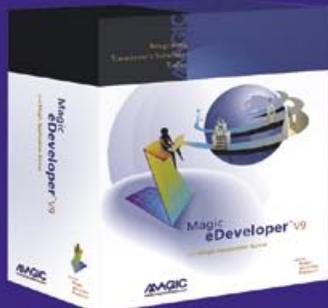
Referencias

- www.ebxml.org



Robusto Orientado a Resultados Inteligente...

MAGIC
eDeveloper™



Magic eDeveloper es la única herramienta con funciones ínter-construidas que facilitan la programación de cualquier desarrollo sin usar una sola Línea de Código.

Magic eDeveloper es abierto ya que se monta en un gran número de plataformas de hardware y sistemas operativos, ofreciendo conectividad directa a la mayoría de las Bases de Datos Relacionales logrando integrar simultáneamente ambientes heterogéneos.

Magic eDeveloper permite liberar rápidamente aplicaciones Web que se pueden integrar fácilmente a plataformas J2EE y/o Web Services.

Magic eDeveloper es una tecnología altamente productiva para el desarrollo dinámico e integración de modernas soluciones a nivel corporativo que le permite administrar y personalizar su negocio de una manera rápida, práctica y sencilla.

Informes:

ROCA Sistemas, S.A. de C.V.

Tel: 5687-6239, 5523-0706

1107-7247, 1107-6899

ventas@rocasistemas.com.mx

www.rocasistemas.com.mx

MAGIC
www.magicsoftware.com

Metodologías Ágiles

EL PROCESO EN LA MEDIDA JUSTA

Por John Gómez

Surgidas principalmente durante los años 90 las metodologías ágiles marcaron un punto de quiebre en la definición y uso de procesos para desarrollo de software. Los conceptos “extremos” y con frecuencia contrapuestos a los establecidos hasta el momento por casi 20 años de estudios en ingeniería de software se extendieron como una inflamación provocando reacciones que fueron desde la idolatría irracional, pasando por el escepticismo prudente, hasta el rechazo categórico y creando una polémica que hoy en día no está zanjada completamente.

Lo que no se discute es que nadie quedó indiferente y hoy son una alternativa real para mejorar los procesos de software. El problema de la polémica es que las posiciones se han radicalizado tanto que las exageraciones y malas interpretaciones están a la orden del día, no sólo por parte de sus detractores sino también de sus promotores.

La idea de este artículo no es resolver la polémica, sino ofrecer algunas herramientas básicas para interpretar correctamente los principios de lo “ágil” de modo que permitan tomar alguna posición propia y descubrir si realmente estos conceptos pueden ser aprovechados en sus proyectos o su organización.

Lo “Ágil” por los “Agilistas”

Para empezar es bueno escuchar lo que dicen los agilistas sobre si mismos y para ello la mejor fuente es el “Manifiesto Ágil”, disponible en www.agilemanifesto.org. Esta es una declaración de los valores y principios que orientan a todas las metodologías ágiles. Fue publicado en 2001 cuando un grupo de creadores y practicantes de las metodologías ágiles más conocidas (XP, SCRUM, Crystal, ASD, FDD, DSDM y Lean Development) notaron que compartían conceptos y visiones y decidieron reunirse para crear un movimiento. Este movimiento se llamó la Alianza Ágil y su primer producto fue este Manifiesto, El mismo término “ágil” se acuñó en esta reunión. Antes de eso las metodologías eran conocidas, a veces despectivamente, como “livianas”.

El texto del Manifiesto dice:

“Estamos descubriendo mejores formas de desarrollar software al hacerlo y al ayudar a otros a hacerlo. A través de este trabajo hemos llegado a valorar:

- A los individuos y sus interacciones por encima de los procesos y herramientas
- Software que funcione por encima de la documentación exhaustiva
- Colaboración con el cliente por encima de la negociación de contratos
- Capacidad de respuesta al cambio por encima de el seguimiento de un plan

Esto significa, que mientras reconocemos que hay valor en los elementos de la derecha, valoramos más los de la izquierda.”

Cada punto del manifiesto tiene una interpretación: así, en el primer párrafo, la palabra “descubriendo” indica que el grupo reconoce que no se tienen todas las respuestas y por otra parte al decir “al hacerlo...” se quiere indicar que esto se realiza a través de la práctica diaria en proyectos y no con base en estudios teóricos.

Después viene la declaración de valores. Cuando se lee resulta evidente por qué se desató la polémica considerando que se estatuyó sobre elementos fundamentales de la ingeniería de software tradicional (mejorar los procesos para mejorar los resultados, fuerte énfasis en la documentación, contratos, planes etc.). Así para los “agilistas” estos elementos son importantes en los

proyectos pero siempre serían menos importantes que las personas y sus interacciones, el producto funcionando, la colaboración con el cliente y la respuesta versátil al cambio. Esto queda explícito en la oración que cierra el manifiesto.

Posteriormente, tenemos los principios que “realizan” los valores y determinan los elementos clave del desarrollo “ágil”. Por falta de espacio no puedo incluir aquí los 12 principios, pero entre los puntos más importantes están los siguientes:

- El objetivo fundamental es la satisfacción de las necesidades del cliente, por lo cual se aceptan los cambios que se consideren necesarios para dar más valor a las soluciones.
- El equipo debe generar un ambiente que permita responder efectivamente a tales cambios.
- El proceso de desarrollo es iterativo e incremental, tratando de entregar al final de cada iteración un producto que el cliente pueda usar, con preferencia por iteraciones cortas (de un par de semanas a un par de meses).
- El equipo es el principal factor de éxito. La gestión del proyecto se orienta a mantener el ambiente adecuado para el equipo, y a favorecer la comunicación frecuente y directa. Al equipo se le da la confianza y el poder de participar en la organización y toma de decisiones. El equipo se hace responsable de reflexionar sobre su desempeño y de promover mejoras a los esquemas de trabajo.

John Gómez es consultor senior en Practia Consulting Chile. Tiene más de 15 años de experiencia en desarrollo de software asumiendo roles de programador, diseñador, arquitecto y jefe de proyecto. Desde el año 2002 colabora con la implementación de prácticas ágiles en proyectos de desarrollo y ha dirigido proyectos de mejora basados en metodologías ágiles. Hoy John está a cargo de varios proyectos de mejora, incluyendo algunos orientados a obtener evaluaciones de CMMI niveles 2 y3.
jgomez@pragmaconsultores.com

ENTRENAMIENTO ACELERADO

Si su empresa está llevando a cabo una transición a la plataforma Java, le ofrecemos un nivel de entrenamiento acelerado de su personal. Evaluando las habilidades de su personal, nos enfocamos únicamente en los aspectos que son específicos a la plataforma Java, aprovechando al máximo las habilidades y conocimientos que ya poseen.

Servicios:

Soluciones Java.

- Taller de Metodología para la Arquitectura
- Desarrollo de Aplicaciones
- Planeación del Proyecto
- Análisis y Desarrollo del Prototipo
- Seminario (Taller de Metodología para la Arquitectura)
- Java Role Analysis (Skill Analysis, Desarrollo Profesional)
- Entrenamiento
- Mentoring & Coaching
- Pre- Certificación
- Sylvan Prometric

Plan de Desarrollo Profesional

- Análisis de Necesidades de Aprendizaje
- Inventario de Análisis de Necesidades
- Capacitación
- Pre-certificación
- Mentoring and Coaching
- Certificación

Análisis de Conocimientos y Habilidades

- Pre- Certificación
- Evaluación Objetiva de Conocimientos (Skill Object)
- Skill Análisis
- Skill Assessment

Estructura Organizacional

- Descripciones, Valuaciones y Perfiles de Puesto
- Sueldos y Salarios
- Detección de Necesidades de Capacitación
- Definición de Estrategias y Desarrollo de Programas
- Cursos, Talleres y Seminarios de Capacitación.

Informes y ventas:

EDUTECSA S.A. DE C.V.

56 51 42 41

56 60 48 06

56 80 39 03

Fax ext. 114

Calz. de los Leones # 117-107A

Col. las Águilas, México, D.F.

01710 Mexico DF.

ventas@edutecsa.com

www.edutecsa.com

- El cliente se convierte en parte del equipo, incluyéndolo también en el círculo de comunicación frecuente y directa a lo largo de todo el proyecto.

La lista completa de principios también está disponible en www.agilemanifesto.org

Entendiendo lo Ágil

Lo “ágil” surge de reconocer por un lado que el desarrollo de software es una labor compleja que requiere de métodos para poderla llevar a buen término y por otro que los métodos que se venían aplicando ya no estaban siendo tan efectivos para lidiar con los nuevos retos a los que se enfrentaban los proyectos. Esto no significa que los métodos fueran malos sino que no eran apropiados y que la causa principal de esta situación es que los métodos eran demasiado “pesados”. Es decir, que el esfuerzo de mantener el proceso resultaba en una carga que hacía más lento el andar del equipo en vez de ser una herramienta en pro de la eficiencia del trabajo. Así se podía ir del extremo de no tener ningún proceso (el clásico caos) a tener un proceso excesivo más allá de las demandas reales del proyecto.

Lo ágil se sitúa en el justo medio, reconociendo la necesidad del proceso pero concediendo espacio al proceso “suficiente”: aquel que permite mantener control del trabajo y hacerlo más eficiente a un costo (esfuerzo) razonable.

Por lo mismo, los procesos ágiles si son “livianos”, ya que tienen por lo general menos prácticas, roles o actividades definidas y porque las definiciones de estos elementos de proceso son menos rigurosas y formales que en los métodos tradicionales. Este bajo formalismo es compensado dentro del mismo método ágil por prácticas de comunicación que aumentan la visibilidad sobre el proceso que se ejecuta. Un ejemplo de esto es la gestión de proyecto en SCRUM: el menor rigor de la planificación es compensado por seguimiento diario de actividades, y problemas y reestimación, también diaria, del esfuerzo pendiente.

Hay dos conclusiones importantes que podemos derivar: una es que el criterio de lo “suficiente” se puede usar como regla general en cada instancia de aplicación de lo ágil. Por ejemplo con la documentación: no se trata de no documentar, sino de documentar lo exclu-

sivamente necesario. La segunda conclusión es que eventualmente las necesidades del proyecto pueden implicar procesos más rigurosos: un diseño liviano levantado en una pizarra al que se le toma una foto digital puede ser insuficiente en el desarrollo de un sistema de control de misiles. El ejemplo puede ser algo extremo pero revela la necesidad de que cada equipo determine el proceso que requiere su proyecto realizando las adaptaciones correspondientes. Esto también se traduce en que algunos de los métodos ágiles de definiciones conocidas (como Extreme Programming, por ejemplo) resulten inaplicables en ciertos tipos de proyecto.

Ampliando las Perspectivas

Hoy existen varias corrientes de evolución de lo ágil en actividad constante. Una destacable es la extensión de los valores, principios y prácticas a disciplinas más allá del desarrollo de software: gestión de proyectos y gestión organizacional por mencionar algunos. Otra muy importante es la que propone que los conceptos derivados de lo ágil y lo tradicional no son contradictorios sino complementarios. En lo personal comparto esta perspectiva y con base en ello hemos desarrollado e implementado métodos que usando prácticas ágiles cumplen con los requerimientos del modelo CMMI. En otra ocasión escribiré al respecto.

Para Profundizar

La mejor fuente para conocer más es el sitio de la alianza ágil (www.agilealliance.com), que tiene muchos vínculos valiosos a artículos, herramientas, eventos y grupos de noticias. Una buena lectura inicial es el artículo “The New Methodology” de Martin Fowler (www.martinfowler.com), uno de los firmantes originales del manifiesto y que contiene muchos vínculos interesantes.

Ya implementando, lo mejor suele ser usar parcial o totalmente algunos de los métodos ágiles conocidos. Mis favoritos y recomendados en ese sentido son SCRUM de Jeff Sutherland y Ken Schwaber y Crystal Clear de Alistair Cockburn. Para los interesados en el balance de métodos, un imperdible es el trabajo de nada menos que Barry Boehm y Richard Turner “Balancing Agility and Discipline”, publicado por Addison Wesley en el 2003. Por cierto, el Dr. Turner es Keynote Speaker invitado a SEPG LA en Guadalajara este año. 

Patrones de Casos de Uso

Por Saúl Cuesta Rodríguez

Al desarrollar o construir algo, como por ejemplo una casa o una máquina, es muy útil apoyarse en la experiencia anterior, ya sea de uno mismo o de otros. De esta manera sabremos que la solución va a funcionar, y tendremos identificados los problemas potenciales, así como soluciones para éstos. Estas soluciones a problemas comunes se conocen como patrones, y se utilizan en diversos aspectos del desarrollo de software. Ya en un artículo anterior de SG se habló sobre patrones de diseño, específicamente del patrón Modelo-Vista-Controlador. En esta ocasión, estudiaremos la aplicación de patrones al modelado de casos de uso, y en específico abordaremos el patrón CRUD.

Contrario a lo que se pudiera pensar, un patrón de casos de uso no describe un uso particular de un sistema. Más bien, captura técnicas para que el modelo sea mantenible, reusable, y entendible. Entonces, podemos decir que los patrones de casos de uso capturan mejores prácticas para modelar casos de uso.

La aplicación de patrones de casos de uso nos trae los siguientes beneficios:

- Aumentar la productividad.
- Reutilizar elementos existentes (en este caso fragmentos de modelos).
- Evitar el retrabajo por errores.
- No invertir tiempo en resolver problemas ya resueltos.
- Aplicar la teoría al trabajo práctico.
- Habilitar las herramientas de soporte para modelar el desarrollo.

Captura y Organización de Patrones

Para capturar y organizar los patrones de caso de uso, recomiendo ampliamente generar un repositorio de patrones donde capturen al menos la siguiente información para cada patrón:

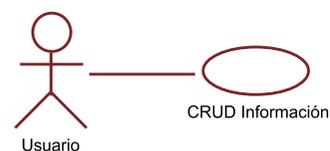
- **Nombre:** el nombre descriptivo del patrón.
- **Intento:** captura cuál es el objetivo de aplicar el patrón.
- **Características:** estados de los patrones, si son simples o complejos, comunes o infrecuentes.
- **Palabras claves:** palabras clave que caractericen al patrón para facilitar su búsqueda.
- **Tipo:** si el patrón afecta la estructura del modelo o la descripción de un caso de uso.
- **Modelo:** un modelo de caso de uso a aplicar.
- **Descripción:** la descripción del modelo.
- **Aplicabilidad:** cuándo y cómo aplicar el patrón.
- **Discusión:** completa discusión sobre el patrón.
- **Ejemplo:** un ejemplo donde uno o más de los patrones se aplica.
- **Modelo de análisis:** diagrama con clases de análisis que proporciona una realización de los casos de uso.

Una vez que hemos hablado sobre lo que es un patrón de caso de uso, y la forma de documentarlos, estudiemos el patrón conocido como CRUD, en sus variantes completa y parcial.

Nombre: CRUD

Intento: este patrón se utiliza en los casos donde se quiere realizar altas, bajas, cambios y consultas a alguna entidad del sistema. Su nombre es un acrónimo de las palabras en inglés Create, Read, Update, Delete. Palabras clave: Creación, altas, bajas, cambios, catálogo

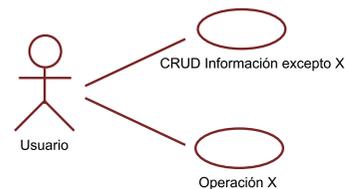
Modelo 1: CRUD Completo



Descripción: el patrón CRUD Completo consiste en un caso de uso para administrar la información (CRUD Información), nos permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y dar de baja.

Aplicabilidad: este patrón deberá ser usado cuando todas las operaciones contribuyen al mismo valor de negocio y todas son cortas y simples.

Modelo 2: CRUD parcial



Descripción: alguna de las alternativas del caso de uso puede ser modelada como caso de uso independiente.

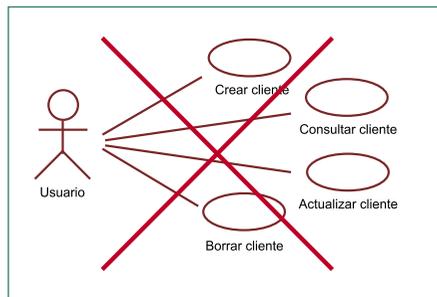
Aplicabilidad: este patrón es preferible cuando uno de los flujos alternativos del caso del uso es más significativo, muy largo, o mucho más complejo que el patrón completo.

Discusión: muy a menudo los sistemas manejan información que, del punto de vista del sistema, se crea muy fácilmente. Después de un chequeo de sintaxis, o quizás un cierto chequeo sin importancia de un cálculo o regla de negocio, la información se almacena simplemente en el sistema. No es necesario realizar cálculos, verificaciones complejas, o recuperación avanzada de datos. La descripción del flujo es pequeña, y probablemente sólo haya una o dos trayectorias alternativas de menor importancia en el flujo. Las consultas, cambios, o bajas son operaciones igualmente simples.

¿Deben tales operaciones ser modeladas como un caso de uso? ¿Y deben incluirse en el modelo del sistema completo? La respuesta a ambas preguntas es sí. Son casos de uso porque son funciones que debe ser capaz de realizar el sistema. Alguien utilizará el sistema para administrar esa información. Por otra parte, deben ser incluidos en el modelo, ya que de otra manera estará incompleto. Y esto puede provocar que el proyecto no se lleve a cabo adecuadamente.

Afortunadamente, esto no significa que necesariamente cada operación se debe expresar como casos de usos separados. Según el patrón CRUD, los podemos agrupar. Este procedimiento tiene algunas ventajas obvias. Primero, el tamaño del modelo será reducido, hará más fácil de entender el modelo porque tiene menos casos. En segundo lugar, nadie estará interesado en un sistema que contiene solamente un subconjunto de estos casos de uso, ya que no generan el valor completo (por ejemplo, leer y dar de baja, pero no crear y cambiar). Agrupar estos flujos juntos en un caso como Administrar X se

asegura de que las cuatro operaciones estarán incluidas en el modelo, y lo hace claro para el lector del modelo. Tercero, el valor de cada uno de los flujos por separado es muy pequeño, y podríamos estar cayendo en descomposición funcional; es la colección entera de estas operaciones la que da valor a los interesados.



Las cuatro operaciones no deben ser modeladas como caso de uso independientes.

La variación denominada CRUD: Parcial, indica que en caso de que solo algunas de las cuatro operaciones sean simples mientras que otras son complejas, se puede agrupar las operaciones simples en un caso de uso y dejar las otras modeladas como un caso de uso separado.

Observar también que esto es una situación típica donde un caso de uso no tiene solamente un flujo “básico”. Ninguno de los flujos se puede decir que es más “básico” o “normal” que los otros. Por lo tanto, un caso de uso CRUD tendrá típicamente cuatro flujos básicos, y posiblemente algunos flujos alternativos, como se demuestra en el ejemplo 1.

Una instancia de un caso de uso podrá realizar una de las siguientes operaciones (crear, leer, cambiar y dar de baja), y después dejar de existir. Esta misma instancia del caso de uso no continuará viviendo y esperando la operación siguiente que se realizará. Esta operación debe ser realizada por otra instancia del mismo caso de uso.



ERP
SS

Integra
Software
Integrador
Empresarial

Contabilidad Dinámica
Reingeniería
Outsourcing
CRM
BI
RUP

Librerías
Universidades
Microcreditos
Gasolineras
Retail
Servicios y
Más...

AnadicSoft México, Guadalajara,
Mexicali, Monterrey, Tabasco,
Tijuana, Chiapas, Distrito Federal.
Culiacan.

ventas@ssistemas.com

01800 - 7706050

www.ssistemas.com

Los patrones de casos de uso capturan mejores prácticas para modelar casos de uso

Como regla general, cuando no estamos seguros si combinar los diversos casos de usos en uno o crearlos como separados, la recomendación es mantenerlos como uno solo y después cuando se vea el tamaño y complejidad del caso de uso, se deberá tomar la decisión de separarlos si es necesario.

A continuación mostramos un ejemplo de un boceto de caso de uso utilizando el patrón CRUD. Por razones de espacio, no está completo. El objetivo es ejemplificar la aplicación del patrón.

Caso de Uso: Administrar Tarea

Este caso de uso permite registrar, listar, modificar, o eliminar la información de tareas realizadas.

Flujo Básico

El caso de uso tiene cuatro diversos flujos básicos:

Registrar tarea nueva
 Modificar una tarea existente
 Cancelar tarea
 Consultar tarea fallida

Registrar tarea nueva

El caso de uso comienza cuando el usuario elige registrar una nueva tarea.

El sistema presenta una lista de posibles clases de tareas, y pregunta qué clase de tarea debe ser registrada, con qué nombre, y cuando se debe realizar.

El usuario provee la información requerida.

El sistema comprueba que el tiempo especificado es en el futuro y que el nombre de la tarea es único.

El sistema registra la nueva tarea y la marca como activa.

El caso de uso termina.

Modificar tarea

El caso de uso comienza cuando el usuario elige modificar una tarea ya registrada.

El sistema recupera los nombres de todas las tareas no marcadas como activo y las presenta al usuario.

El usuario selecciona una de las tareas.

El sistema recupera la información sobre la tarea y la presenta al usuario.

El usuario modifica cualquiera de la información actual excepto el nombre de la tarea.

El usuario acepta la información.

El sistema comprueba si el tiempo especificado es en el futuro y almacena la información modificada.

El caso de caso de uso termina.

Cancelar tarea

El caso de uso comienza cuando el usuario elige cancelar una tarea.

El sistema recupera y despliega todas las tareas futuras.

El usuario selecciona una de las tareas.

El sistema recupera la información sobre la tarea y la presenta al usuario.

El usuario confirma la cancelación, el sistema elimina la tarea; si no se puede, no se hace ninguna modificación.

El caso de uso termina.

Consultar tarea fallida

El caso de uso comienza cuando el usuario elige vista de la lista de todas las tareas que han fallado.

El sistema recoge todas las tareas con el estado fallado y presenta sus nombres al usuario.

El caso de uso termina.

Flujos alternativos

Cancelar la operación
 Nombre o Tiempo incorrecto

Otros Patrones de Caso de Uso

Existen varios patrones adicionales al CRUD que son utilizados para modelar sistemas. Esta es una lista de los más importantes:

- Reglas de negocio
- CRUD
- Login
- Reportes y Explotación de información
- Inclusión y Extensión
- Sistema en Capas
- Múltiples Actores
- Jerarquía de Componentes
- Servicios Opcionales

Conclusión

La técnica de casos de uso nos permite modelar y especificar los requerimientos de nuestro sistema. Tiene muchos beneficios, entre los más importantes: primero nos permite planear nuestro proyecto y segundo ayuda a llegar a acuerdos con los usuarios, para que los casos de uso sean mas claros y mantenibles es importante encontrar patrones y documentarlos, de esta manera cuando nos encontremos con un problema igual o parecido podamos resolverlos en menor tiempo. El concepto de patrones no es algo que solo es aplicable a la práctica de requerimientos. De hecho, la disciplina de requerimientos copia este concepto de la de análisis y diseño. Lo que se busca con los patrones es reutilizar lo aprendido en los nuevos proyectos y usarlos en la organización como estándares. 

Referencias

- Pan-Wei Ng, *Understanding types of use cases and artifacts*. IBM Developerworks. www-128.ibm.com/developerworks/rational/library/1809.html

LINQ

REDUCIENDO LA COMPLEJIDAD DEL ACCESO A DATOS

¿Dónde se encuentra la mayor complejidad en el desarrollo de aplicaciones actuales? Definitivamente son muchas las áreas donde se continúan requiriendo avances importantes:

- Ingeniería de software, modelado y verificación.
- Implementación, lenguajes de programación y plataformas tecnológicas.
- Artefactos, herramientas y sus relaciones para crear una arquitectura de software.

El constante cambio de paradigmas impide resolver problemática tradicional de construcción de software. Por ejemplo, el cambio del **software como servicio**, requiere evolucionar de “objetos” a “servicios”. En su artículo “Introducing Indigo: An Early Look”^[1], David Chappell explica claramente nuevos fundamentos que simplifican dramáticamente la construcción de dicho sistema: las transacciones distribuidas son transparentes al programador, labor que actualmente es altamente compleja con arquitecturas orientadas a servicios.

Luego de dos décadas, la industria ha alcanzado una madurez en la capacidad de crear aplicaciones orientadas a objetos y componentes. Pero hoy, sin importar el modelo de construcción de software, se ha hecho claro que el gran reto en tecnología de programación es indudablemente reducir la complejidad de acceso a fuentes no orientadas a objetos —particularmente bases de datos relacionales y XML.

LINQ

Language INtegrated Query (LINQ) es una propuesta para simplificar el acceso a bases de datos que será incorporada por Microsoft en los próximos meses. Esta tecnología es posible gracias a la explotación de metadatos para efectuar análisis en tiempo de diseño. Hoy en día, el interactuar con una base de datos relacional es “trabajar a ciegas” desde el código fuente porque es una entidad totalmente aislada.

LINQ define un conjunto de operadores estándar que pueden ser usados para recorrer, filtrar y proyectar operaciones en forma declarativa. Sobre XML los operadores son eficientes en consumo de recursos de cómputo y ofrecen funcionalidad XPath/XQuery al lenguaje de programación.

LINQ son extensiones a VisualBasic y forman parte de C# 3.0. Por ejemplo, si existe una colección Países, se pueden encontrar aquellos con menos de un millón de habitantes en forma muy parecida a Structured Query Language (SQL):

```
Dim PaísesChicos = Select Países _
From Pais In Países _
Where Pais.Habitantes < 1000000
```

Lo grandioso del anterior ejemplo es que puede operar igualmente sobre objetos, XML o una base de datos relacional. Los desarrolladores ya no tendrán que manipular dos mundos separados con vocabularios propios en forma ineficiente. La reducción en código escrita será significativa. Por ejemplo, el siguiente pedazo de código:

```
XmlDocument doc = new XmlDocument();
XmlElement contactos = doc.CreateElement("contactos");
foreach (Cliente c in Clientes)
    if (c.Pais == "MEXICO") {
        XmlElement e = doc.CreateElement("contacto");
        XmlElement nombre =
            doc.CreateElement("nombre");
        nombre.InnerText = c.CompanyNombre;
        e.AppendChild(nombre);
        XmlElement telefono =
            doc.CreateElement("telefono");
        telefono.InnerText = c.Telefono;
        e.AppendChild(telefono);
        contactos.AppendChild(e);
    }
doc.AppendChild(contactos);
```

Con LINQ se escribe:

```
XElement contactos = new XElement("contactos",
    from c in Clientes where c.Pais == "MEXICO"
    select new XElement("contacto",
        new XElement("nombre", c.Nombre),
        new XElement("telefono", c.Telefono) ) );
```

LINQ es un esfuerzo por unir a los lenguajes de programación el acceso a datos universal. No es una librería, API o sistema de administración de base de datos en memoria. Se construye sobre la extensibilidad de los lenguajes de programación explotando fuertemente los metadatos disponibles.

Para los desarrolladores, el sueño más importante de productividad es contar con un modelo único de programación. Esencialmente, LINQ permitirá acceder datos nativamente utilizando la sintaxis de programación de su lenguaje preferido.

Para mayor información sobre LINQ, visiten msdn.microsoft.com/netframework/future/linq/

-Luis Daniel Soto



Luis Daniel Soto Maldonado es Director de Evangelización en Nuevas Tecnologías en Microsoft México. Entre sus funciones actuales están la administración de la relación con el Gobierno Mexicano para el desarrollo de la industria de software (ProSoft). Luis Daniel es jurado del “Gran Orden de Honor al Mérito Autoral” en software del INDAUTOR/SEP y fundador de diversas asociaciones de Tecnologías de Información (TI) relacionadas a inteligencia competitiva, administración del conocimiento y construcción de software. Luis Daniel Soto es Ingeniero en Sistemas de la Fundación Arturo Rosenblueth y ganó el primer lugar en el concurso nacional para software de exportación en 1989. blogs.msdn.com/luis-dans

Referencias

1. www.davidchappell.com/articles

Modelo Conceptual

DOMINANDO EL PROBLEMA

Por Sergio Orozco

Sin duda, una de las preguntas más frecuentes hacia quienes capacitamos en UML, es: ¿cuáles son los artefactos mínimos indispensables para obtener beneficios tangibles en los proyectos de software? Es difícil expresar en un artículo de este tamaño la respuesta a esta pregunta, aunque trataré de simplificar la respuesta que normalmente damos.

Por mi experiencia implantando procesos centrados en UML, puedo asegurar lo que ya muchos saben: en esto de los procesos el mundo no es color de rosa. Mientras que algunos puristas podrían sugerir usar la mayoría de los artefactos en cada proyecto, la verdad de las cosas es que en muchas ocasiones esto no es factible. En los proyectos de hoy en día, difícilmente se cuenta con tiempo suficiente para modelar todos los aspectos de un sistema, es por ello la eterna búsqueda del mínimo indispensable.

Yo creo en la filosofía de “es mejor poco que nada”, pues he aprendido en todos estos años que sugerir el uso de todos los artefactos ocasiona que al final ninguno se utilice, se use el menos apropiado o se hagan inadecuadamente. Así que mi recomendación es usar por lo menos el diagrama de casos de uso y/o el diagrama de clases, con el primero para obtener más beneficios en cuanto a calidad y control del proyecto, y el segundo para desarrollar sistemas más orientados a objetos. Dado que en números anteriores ya hemos hablado sobre casos de uso, en esta ocasión me concentraré en los diagramas de clases.

¿Uno o Dos Artefactos?

Posiblemente se sientan tentados, o estén acostumbrados a desarrollar el diagrama de clases desde una sola perspectiva, modelando las clases de software a implementar. Es decir, realizando directamente el diseño de la aplicación. Mi recomendación al respecto es que si quieren sacarle el máximo provecho al diagrama de clases es conveniente desarrollarlo en dos ciclos; Uno de análisis y otro de diseño, lo cual implica que en realidad están desarrollando dos artefactos en el proceso en lugar de uno, ambos usando el diagrama de clases como base.

En el primero de estos ciclos, el de análisis, se desarrolla lo que podemos llamar diagrama preliminar de clases, modelo del dominio o modelo conceptual. No importa el nombre, lo que importa es lograr el objetivo de comprender el dominio (contexto del problema); con el beneficio indirecto de estar estableciendo las posibles clases del sistema. Ya en el ciclo de diseño este diagrama preliminar será usado como una base a refinar para determinar las clases definitivas a implementar en el sistema orientado a objetos.

La Comprensión del Problema

En este artículo nos enfocamos en la versión de análisis del diagrama, conocido como modelo conceptual. Su relevancia radica en que si no comprendemos el dominio del problema y las reglas de negocio, habrá pocas esperanzas de sugerir y desarrollarle un buen sistema a nuestro cliente. Y entre más complejas sean las reglas de negocio, más fácilmente tenderemos al fracaso si no logramos esta comprensión.

Imaginen que les solicitaron desarrollar un sistema para vender pólizas de seguros de vida. Pero, quizás nunca han comprado un seguro, por lo que no tienen idea de los conceptos asociados. No saben que las pólizas aseguran a un cliente por un monto determinado, ni que existen diferentes tipos de planes de acuerdo a las características, ni que el cliente puede seleccionar diferentes planes de pago, ni tampoco que la póliza puede tener desde uno hasta un máximo de diez beneficiarios. Bueno, pues si yo fuera el cliente y ustedes no logran comprender los puntos anteriores, tengan por seguro que buscaría a alguien más que lo desarrollara. Y este es un ejemplo simple, pues sabemos que las reglas de negocio de cualquier sistema pueden ser mucho más complejas. No por nada alguien dijo: “empiezo a pensar que es más fácil enseñarle a mi gente de negocios cómo desarrollar sistemas que a la gente de sistemas el negocio”.

Enlistar textualmente estas reglas en un documento puede ser útil, pero cuando tienes un documento de varias hojas para explicar el dominio es muy fácil que la gente comience a sentirse agobiada por tanta información.

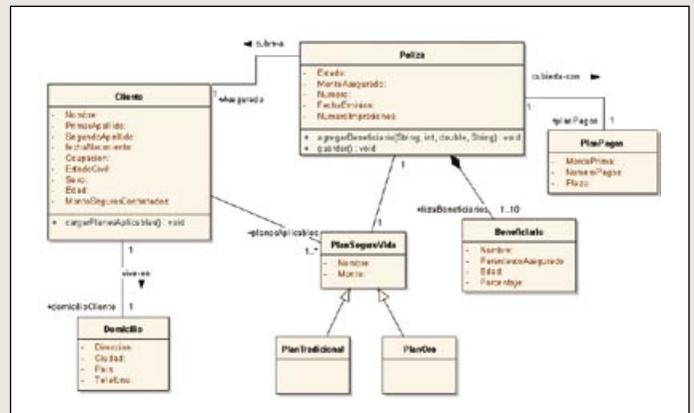


Figura 1. Modelo conceptual para la venta de seguros de vida.

Sergio Orozco es director general e instructor senior certificado por la OMG en Milestone Consulting, empresa especializada en capacitación práctica y consultoría en UML, CMM y orientación a objetos. Milestone Consulting es la primer empresa mexicana miembro de la OMG. info@milestone.com.mx www.milestone.com.mx

En cambio, si usas un modelo conceptual para expresarlas, será mucho más fácil documentarlas, analizarlas y comprenderlas. Con la ventaja, como ya comenté, que estarás estableciendo las bases para construir una aplicación orientada a objetos.

Los elementos principales a mostrar en el modelo conceptual son:

- **Conceptos.** Elemento lógico o físico que ayuda a entender el problema, es parte del lenguaje utilizado por el cliente y generalmente se nombra como sustantivo. Se representan con el símbolo de una clase. Ejemplo: Cliente, Póliza y Domicilio.
- **Atributos.** Información que caracteriza al concepto en el mundo real. Se muestra en el segundo compartimiento de las clases. Ejemplo: Nombre, apellidos y edad del cliente.
- **Asociaciones.** Relaciones lógicas o físicas que existen en el mundo real entre dos conceptos. Si puedes armar una frase con dos conceptos significa que la puedes representar mediante una relación de asociación en-

tre esos dos conceptos. Puedes colocarle el verbo que usas para relacionar los conceptos en la frase, indicándolo sobre la asociación con una punta de una flecha para indicar la dirección en que se debe leer la frase. Ejemplo: La Póliza cubre-a un cliente asegurado, el cliente vive-en un domicilio.

- **Rol.** El rol también puede aclarar la relación entre dos conceptos, indica el rol que juega un concepto con respecto a otro en una relación de asociación. Ejemplo: Planes- Aplicables al cliente.
- **Multiplidad.** El número de instancias de un concepto relacionados con el otro concepto. Ejemplo: Una póliza tiene una lista de uno a diez beneficiarios.
- **Generalización.** En lugar de poner una asociación para armar la frase “es-un-tipo-de”, utilizamos una generalización. Esto puede llegar a confundir al lector del modelo, por lo que hay que asegurarse que entienda perfectamente el significado de la notación. Ejemplo: El Plan Oro es un tipo de plan de seguro de vida, al igual que el plan tradicional.

• **Agregación y composición.** Indican una relación donde uno de los conceptos es el contenedor del otro. Ejemplo: la póliza contiene una ListaBeneficiarios.

Estos son los elementos básicos a utilizar para aclarar el dominio de un problema. Algunas personas gustan incluir las operaciones que reflejan el comportamiento o acciones que puede realizar cada elemento. Ejemplo: a un cliente se le pueden cargarPlanesAplicables(). Yo generalmente prefiero definir las operaciones durante una actividad separada de diseño, pero si te da resultado de otra forma está bien.

Un tip adicional, aunque este diagrama se parece a un diagrama entidad relación, no trates de modelarlo siguiendo las reglas de ese tipo de diseño. Un entidad relación es un modelo orientado a implementar una base de datos relacional, y el modelo conceptual como lo dice el nombre, sólo muestra conceptualmente el dominio del problema. ☺

PERVASIVE

La Base de Datos Cliente/Servidor con el menor costo total de propiedad del mercado, disponible para plataformas NetWare, Windows (2003/2000/NT/XP/Me/98) y Linux (2.4 Kernel o Better).

Una mezcla perfecta entre precio, versatilidad, desempeño y facilidad de administración

gopacHE
HERRAMIENTAS PROFESIONALES PARA TI

¡NUEVO!
PERVASIVE
PSQL v9™
Experience the Power of Embedded



DISTRIBUIDORES DE LAS MEJORES HERRAMIENTAS
PARA PROFESIONALES DE TI

ventas@gopac.com.mx 01 800 87GOPAC

Guadalajara: Tel:(33) 3122 1947, Fax:(33)3121 4984, Av. Cubilete No. 800-11, Col. Chapalita c.p. 45040, Zapopan, Jalisco, México
Cd. de México: Tel:(55) 5601 7343, Fax:(55) 5601 4424, San Lorenzo No. 1009, 4º piso, Col. del Valle, c.p. 03100, México D.F.

www.gopac.com.mx/pervasive

El Proceso de la Prueba de Software

LENGUAJES DE DEFINICIÓN DE PROCESOS

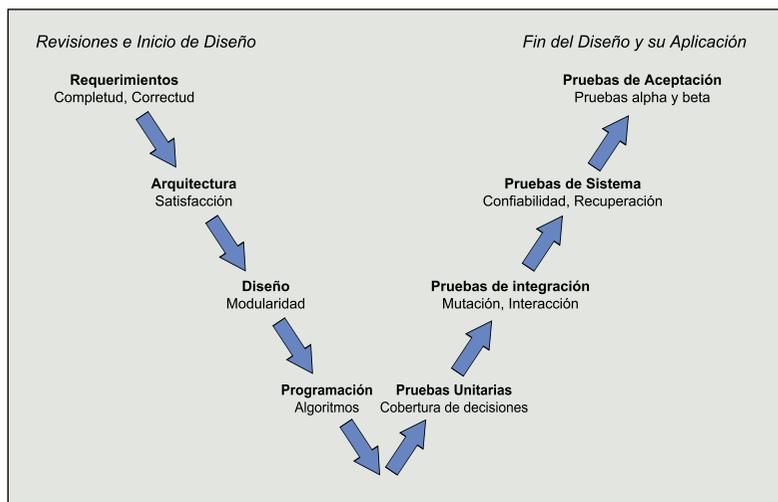


Luis Vinicio León Carrillo es profesor-investigador del Departamento de Electrónica, Sistemas e Informática del ITE-SO, y director general de e-Quality S.A. de C.V., empresa especializada en prueba de software. Luis Vinicio es doctorando por la Universidad Técnica de Clausthal, Alemania. Luis Vinicio es coautor de un marco tecnológico que hoy permite a e-Quality desarrollar empresas de prueba de software. En el contexto de ese marco, su tesis doctoral tiene que ver con aplicaciones de métodos y lenguajes formales que para hacer más eficiente y efectiva la prueba de software. Luis Vinicio es co-fundador del Capítulo Guadalajara de la AMCIS y su Secretario actual.

En este número y el siguiente abordaremos el proceso de la prueba de software. Iniciaremos con el tratamiento formal del concepto, para luego pasar a su aplicación, vinculándolo con la administración de proyectos.

Una Definición Formal

Es común encontrarse en la literatura que aborda el proceso de prueba una referencia al “Modelo-V”. Mediante este modelo se describe a un nivel muy alto de abstracción las fases del ciclo de desarrollo en las que (idealmente) se involucra la prueba; La siguiente gráfica ilustra una adaptación de este modelo, que incluye algunas actividades de cada fase.



Este modelo tiene la ventaja de ser bastante intuitivo: la prueba comienza haciendo revisiones técnicas a los requerimientos y bosquejando los primeros casos de prueba de aceptación, pasando luego a revisar que la arquitectura satisfaga los requerimientos y a definir los primeros casos de prueba de sistema; después se revisa la modularidad del diseño y se bosquejan casos de prueba de integración, para luego pasar a revisar los algoritmos y a desarrollar los casos de prueba de unidad.

Las actividades de prueba de la línea izquierda de la “V” se llevan a cabo en paralelo al desarrollo de software e involucran también la revisión de apego a estándares; las de la línea derecha involucran la terminación del diseño de los casos de prueba y la aplicación de los mismos.

Una desventaja del modelo es que requiere aún de mucho detalle para ser útil en la práctica. Además, la documentación de procesos es una actividad en sí misma, y los documentos generados suelen ser muy propensos a quedar rápidamente inconsistentes (a causa de los cambios) y/o sin actualizar (por la dificultad para realizar esos cambios).

Osterweil^[1]. De manera semejante a lo que ocurre con los BPMs, el utilizar un PDL para definir un proceso de prueba posibilita la generación de un sistema de información que nos permite integrar herramientas CAST y administrar proyectos de prueba de software.

En la empresa e-Quality hemos desarrollado un PDL para definir nuestros procesos y hemos observado que facilita mucho las actividades de diseño, documentación, análisis, mejora y mantenimiento de los mismos, debido a que la documentación adquiere muchos de los atributos de un lenguaje de computación, entre otros: es posible escribir descripciones más precisas y sucintas, con menos probabilidad de contener ambigüedades; las descripciones pueden modularizarse con facilidad, posibilitando el reuso y el mantenimiento efectivo.

Aquí presentaré un PDL mucho más sencillo, en el que se omiten algunas cosas por cuestiones de espacio, pero tratando de incluir las ideas esenciales. Por la misma razón de espacio, no utilizaré grafos de sintaxis, sino que haré la definición utilizando BNF, con los terminales en negritas ^[2] y “λ” denotando la cadena nula; los no-terminales marcados con itálicas entre signos de mayor que y menor que (<Así>), no son definidos con mayor detalle; además no especificaré — entre otras cosas — lo que tiene que ver con el sistema de tipos (básicos, constructores, constantes y variables) ni los mecanismos de paso de parámetros. El PDL es más bien procedural y permite definir procesos dentro de otros procesos (tipo Pascal), posibilitando la generación de una jerarquía de procesos; los niveles permitirían diferenciar entre procesos y procedimientos.

```

Documentación_del_Proceso →
  proceso Identificador() ;
  <Historia_de_Cambios> ;
  Cuerpo
Cuerpo → <Constantes_Tipos_Variables>
  Subprocesos
  Secuencia_de_Actividades
Subprocesos → proceso Identificador() ;
  <Historia_de_Cambios> ;
  Cuerpo ; Resto1
Resto1 → Subprocesos | λ
Secuencia_de_Actividades →
  inicio Actividades fin
Actividades → Actividad Resto2
Resto2 → Actividades | λ
Actividad → <Definición_de_Tarea>
  | Llamada_a_subproceso
  | Secuencia_de_Actividades
  | Alternación_de_Actividades
  | Repetición_de_Actividades
  | Paralelización_de_Actividades
  | λ
Llamada_a_subproceso → Identificador() ;
Alternación_de_Actividades →
  si (<Condición>) Actividad Resto3 fin_s
Resto3 → sino Actividad | λ
Repetición_de_Actividades →
  mientras ( <Condición> ) Actividad fin_m
Paralelización_de_Actividades →
  paralelizar Secuencias_de_Actividades fin_p
Secuencias_de_Actividades →
  Secuencias_de_Actividades Resto4
Resto4 → Secuencias_de_Actividades | λ

```

En el siguiente número utilizaremos este PDL para documentar formalmente una sección de un proceso de prueba, y lo vincularemos con las fases de la administración de proyectos propuestas por el Project Management Institute. ©

- Luis Vinicio León Carrillo

Referencia

1. Osterweil, L. "Software Processes are Software too". Proceedings of the ninth International Conference on Software Engineering. Marzo, 1987
2. Aho, Ullman & Sethi. Compiladores: Principios, Técnicas y Herramientas. Addison Wesley, 2000
3. BNF and EBNF: What are they and how do they work?

www.garshol.priv.no/download/text/bnf.html



www.tcds.com.mx

Evoluciona ágilmente.

Logra consistentemente tu propósito en un mundo complejo de cambios constantes.



Orquestación y Automatización de Negocios | **BOA**

Administración de Procesos de Negocio | **BPM**

Mapeo de Arquitecturas de Procesos | **MAP**

Calidad de Software | **SwQ**

Ingeniería Unificada de Negocios | **UBE**

Administración Integral de Proyectos | **IPM**

Administración del Desempeño Organizacional | **CPM**

Orientación a Objetos

PRINCIPIOS BÁSICOS

Hace ya décadas que el paradigma de orientación a objetos (OO) tomó su lugar en el desarrollo de software. Sin embargo, a la fecha muchos de nosotros seguimos sin entender o aplicar correctamente los conceptos en los que se basa este paradigma, así que hemos decidido aprovechar este espacio, ya sea para conocerlos, repararlos o aclararlos.

Los conceptos fundamentales que se deben entender para poder ser buenos desarrolladores OO, son los siguientes: abstracción, encapsulación, identidad, modularidad, jerarquía, tipos, concurrencia y persistencia. Nótese que estos conceptos no son exclusivos de la orientación a objetos, simplemente tienen ciertas particularidades desde esta perspectiva. Veamos cuales son.



Abstracción

El paradigma OO se basa en la noción de representar elementos del mundo real como objetos. Sin embargo, cualquier elemento del mundo real tiene una cantidad interminable de propiedades y comportamiento. Para lidiar con esta complejidad, utilizamos la abstracción. La abstracción es el mecanismo a través del cual *nos enfocamos en los aspectos esenciales o distintivos de algo, ignorando detalles irrelevantes*. Obviamente, la abstracción siempre se hace desde alguna perspectiva particular, porque, lo que en algunos casos es irrelevante, en otros no es necesariamente así.

Los objetos que encontramos que tienen la misma estructura, comportamiento y semántica, forman una clase. Así que lo que un objeto puede saber (estado) o hacer (comportamiento), está determinado por la clase a la que pertenece.

Identidad

La identidad es la propiedad de un objeto, que lo distingue de todos los demás. Los seres humanos tenemos huellas digitales, números de identificación, perfiles DNA. Estos representan el hecho de que cada uno de nosotros es único e identificable. De la misma manera, cada objeto en un sistema OO tiene una identidad distinta. La identidad es necesaria para que podamos hablar con un objeto sin confundirlo con otro, y para que puedan existir al mismo tiempo varios objetos de la misma clase.

En ocasiones se tiende a confundir la identidad de un objeto con su estado. El estado es el conjunto de valores que encapsula un objeto. Dos objetos pueden tener estados idénticos, pero siguen siendo objetos separados, distintos e identificables.

Encapsulación

Los detalles de una clase —estructuras de datos, algoritmos, etc.— se hacen privados, o encapsulan, para que sea imposible que otras clases dependan de ellos. El principal beneficio es que se una clase pueda modificar la forma en que realiza una operación, sin necesidad de afectar a sus clientes.

La forma en que esto se logra, es separando cada clase en dos partes: su interfaz (qué es lo que hace), y su implementación (cómo lo hace).

Modularidad

La modularidad consiste en la descomposición de algo grande y complejo, en partes más sencillas y manejables.

Mientras que la abstracción se enfoca en reducir la complejidad lógica, la modularidad se preocupa por aspectos físicos o de implementación. Por ejemplo, las clases se agrupan en paquetes para poder administrarlas mejor.

Jerarquía

Una jerarquía es una organización de elementos de acuerdo a su tipo, de acuerdo a una estructura de árbol. Así como en la botánica se utilizan jerarquías para definir familias de plantas, en OO, las jerarquías facilitan reconocer similitudes y diferencias entre objetos.

Los dos tipos de jerarquías más comunes en OO son la jerarquía por herencia o generalización, y la jerarquía por agregación. En la primera, se aplica la frase “es un tipo de”, mientras que en la segunda se aplica “es parte de”. Por ejemplo, una manzana es un tipo de fruta, y a su vez puede ser parte de una cosecha.

Tipos

La importancia de los tipos varía dependiendo de si un lenguaje es estricto con los tipos (*strongly typed*), o no. En los lenguajes OO, los tipos normalmente se refieren a las clases.

En los lenguajes “fuertemente tipificados”, el compilador puede detectar cuando se está tratando de enviar a un objeto, un mensaje que no puede o no sabe responder. Esto evita errores en tiempo de ejecución, además de permitir una mejor optimización del código ejecutable. La desventaja es una menor flexibilidad durante el desarrollo.

Concurrencia

La concurrencia se preocupa por administrar el acceso a recursos compartidos entre operaciones que se superponen en el tiempo (incluyendo la ejecución en paralelo).

Supongamos que tenemos un proceso con múltiples hilos de control. Es posible que un objeto reciba un mensaje al mismo tiempo (aproximadamente) de dos objetos diferentes. Este es un escenario que debemos considerar y manejar apropiadamente.

Existen dos estrategias básicas de control de concurrencia: pesimista, optimista y muy optimista.

En el control pesimista, cuando un objeto inicia acceso a un recurso compartido, le pone un candado, realiza el trabajo que necesita, y una vez que termina, libera el recurso. Esta estrategia no es muy escalable, por lo que sólo se debe aplicar en sistemas pequeños o donde el acceso a recursos compartidos sea raro.

El control optimista utiliza un acercamiento más complejo, pero más escalable. Cuando un objeto inicia el acceso a un recurso compartido, le pone una "marca" única a éste. Después procede a realizar su trabajo, y cuando es momento de aplicar los cambios necesarios en el objeto

compartido, le pone un candado, verifica que su marca no haya sido modificada por otro objeto, aplica los cambios, y libera el candado. En dado caso que la marca haya sido modificada, se dice que hay una colisión. Las colisiones se pueden manejar de diferentes maneras, desde ignorarlas, hasta avisar al usuario de la aplicación para que elija la acción deseada.

Persistencia

Los objetos tienen un periodo de existencia, desde los más volátiles, hasta los más estables. Si un objeto requiere sobrevivir al proceso en que se ejecuta, entonces se dice que es persistente. En otras palabras, la persistencia es la capacidad de un objeto para existir más allá del proceso que lo ejecuta.

Para implementar la persistencia, requerimos de algún mecanismo para almacenar datos. Existen diversos mecanismos, desde archivos en texto plano, hasta bases de datos relacionales (RDBMS) orientadas a objetos (OODBMS).

Con esto concluimos nuestra revisión sobre los principios básicos de la orientación a objetos. Recuerden que para ser un buen desarrollador, se necesita contar con una buena base teórica. ©

Referencias

- Atomic Object. *Object Oriented Tutorial*. atomicobject.com/training-material.page



↳ Capacitación especializada en desarrollo de software

**CURSOS
SEMINARIOS
TALLERES
DIPLOMADOS**

**RENTA DE SALAS
DE CAPACITACIÓN**

**JAVA
.NET
UML
PMI
BPM
PAQUETERÍA**

↳ Consultoría en soluciones de software comprometida en alinear la tecnología con la visión de su negocio utilizando herramientas de vanguardia.

↳ Arquitecturas .NET, J2EE y aplicaciones móviles

↳ Nuestro trabajo de Desarrollo de Software tiene como marco de referencia el modelo Internacional CMMi-3

visionconsulting.com.mx
5524.1032 informes@visionconsulting.com.mx



Los socios OIMA y UNLS son miembros asociados del
Center Management Group, Inc. en la Oficina Central de América y en otros países.

visiontraining.com.mx
5605.6004 capacita@visiontraining.com.mx

visionsoftwarefactory
vision consulting

↳ La verdadera fabrica de Software que entrega productos de Software en tiempo, confiabilidad, costo predecible y administrable.

vswf.com.mx
(442)245.1153 (442)245.1082 Querétaro Qro.

Arreglos de Discos

QUÉ SON Y DÓNDE UTILIZARLOS

Por Ariel García

Un arreglo redundante de discos independientes (RAID por sus siglas en inglés) es típicamente implementado para la protección de la información o incremento del desempeño al acceso de los discos duros. Existen varios tipos de arreglos y los más usados en la industria son: 0, 1, 5 y el 0+1 ó 10, siendo este último el de mayor desempeño, protección y costo.

Actualmente prevalece el uso de este tipo de configuraciones para la protección de la información, pero la industria del almacenamiento y las aplicaciones están evolucionando. Al día de hoy el fabricante de software o la industria del hardware nos entregan soluciones que nos permiten despreocuparnos en cierto grado en la definición del tipo de arreglo a utilizar o para qué archivos en específico los necesitamos. Obviamente estas soluciones son las de mayor costo y mejor desempeño.

Desafortunadamente, sólo aquellas empresas con enormes presupuestos para IT pueden adquirir soluciones de este tipo. En este artículo se explicará cómo los arreglos de disco nos ayudan no sólo a proteger la información, sino también a incrementar el desempeño de nuestras aplicaciones. Primero daremos una breve explicación de qué son los arreglos de discos y los tipos más populares a implementar. Después daremos una recomendación para la configuración de arreglos en las base de datos más comunes que podemos encontrar en el mercado.

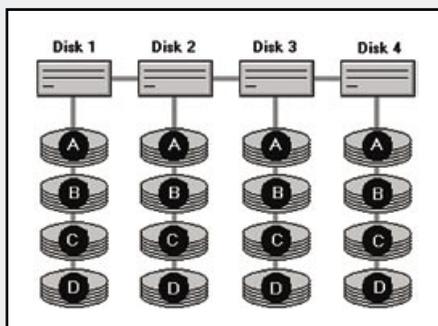
¿Qué son los Arreglos de Discos RAID?

RAID proviene del acrónimo del inglés "Redundant Array of Independent Disks", que significa matriz redundante de discos independientes. RAID es un método de combinación de varios discos duros para formar una unidad lógica única en la que se almacenan los datos de forma redundante. Ofrece mayor tolerancia a fallos y más altos niveles de rendimiento que un sólo disco duro o un grupo de discos duros independientes.

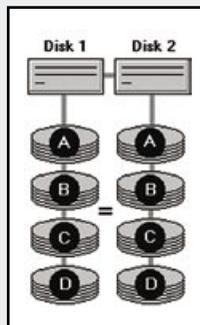
Tipos de Arreglo de Disco

RAID0. Este arreglo es conocido como distribuido (striping), porque utiliza un sistema que

utiliza a los discos como uno solo, teniendo un conjunto de cabezas independientes para su uso. La información es dividida en bloques de datos que se distribuyen en todos los discos del arreglo. EL RAID0 incrementa el desempeño, la lectura y escritura de la información al escribir un solo dato con varias cabezas de forma simultánea. Ejemplo: un dato de 8 bits se divide en todos los discos escribiendo 2 bits en cada uno de forma simultánea. Esto es más rápido que escribir 8 bits de forma serial con una sola cabeza. Este tipo de arreglo no tiene nivel de protección. En caso de la falla de un disco, se perdería toda la información.

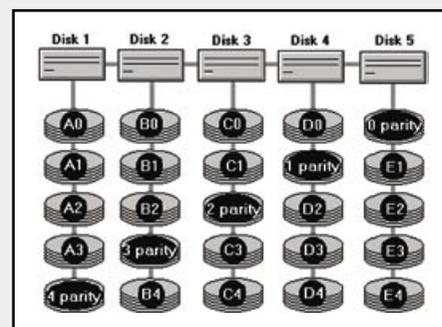


RAID1. Este tipo de arreglo se conoce como Espejeo (Mirroring), porque su conjunto de discos los utiliza como espejos. Ofrece el nivel de protección más alto, pues uno tiene copia idéntica de la información de cada disco. Toda la información escrita en el disco primario se escribe en el disco secundario. RAID1 tiene un incremento en el desempeño de la lectura de la



información, pero puede llegar a degradar el desempeño de la escritura.

RAID5. Este tipo de arreglo se denomina también como distribuido con paridad. Este tipo de arreglos distribuye la información en todo el conjunto de discos. A diferencia del RAID0, RAID5 elabora un bit de paridad con el cual es posible reconstruir la información del arreglo en caso de la pérdida de alguno de los discos. La información y los bits de paridad son distribuidos en todos los discos, garantizando que siempre se encontrarán en discos distintos. RAID5 tiene un mejor desempeño que RAID1, pero cuando uno de los discos falla, el desempeño de la lectura llega a degradarse.



Raid10 (0+1). Este tipo de arreglo es una mezcla del arreglo distribuido y espejeo. La información se distribuye en un conjunto de discos como un RAID0 y, a su vez, este conjunto de discos es espejeado a otro conjunto de discos como un RAID1. RAID10 provee el nivel de protección y desempeño más alto para escritura y lectura que cualquier otro arreglo, debido a que contiene los beneficios de los arreglos distribuidos y espejo. Su único problema es el costo de implementación, al tener que usar siempre el doble discos.

NUESTRA MISIÓN CREAR SU SOLUCIÓN

GRUPO DE SERVICIOS DE TECNOLOGÍAS DE INFORMACIÓN, S.A. DE C.V.



- ▶ Desarrollo de Aplicaciones.
(Expertos en Tecnología .Net y Java)
- ▶ Servidores de Aplicaciones.
- ▶ Administración de Bases de Datos.
(Expertos en Oracle y SQL Server)
- ▶ Tecnología Multimedia.
- ▶ Transferencia de Mejores Prácticas.



Retorno Vía 5 No. 106, Tabasco 2000,
C.P. 86035, Villahermosa, Tabasco.
TELS.: (993) 317 6170 y 71
LADA SIN COSTO: 01800 505 4041
www.gsti.com.mx

ORACLE
PARTNER NETWORK

Microsoft
CERTIFIED
Partner

¿Dónde Puedo Utilizar un Arreglo de Disco?

Ahora que recordamos cuáles son los principales arreglos de disco, podemos revisar dónde podemos utilizarlos. Como tal, no existe una regla para definir dónde deberíamos o no utilizarlos. Esto generalmente depende de nuestros presupuestos y criticidad de los sistemas. En esta ocasión daremos una recomendación para las bases de datos más usadas en el mercado: Oracle, MS SQL y Exchange.

Ninguna de estas recomendaciones está labrada en piedra, la experiencia y el conocimiento de nuestros sistemas nos orientarán hacia cual es nuestra configuración óptima.

ORACLE DATABASE SERVER		Comentarios
Tipo de archivo	Arreglo recomendado	
Control Files	NO RAID, RAID0, RAID 0+1	Los archivos de tipo Control Files son actualizados constantemente, por ello, entre más rápido puedan ser accedidos, mejor. Se recomienda utilizar múltiples Control Sets se tengan o no configurados en un arreglo de discos. En caso de usar NO-RAID o RAID0, asegúrenos tener al menos una copia de los Control Sets siempre disponible.
Online Redo Logs	NO RAID, RAID0, RAID 0+1	Sus necesidades de acceso son similares a los Control Files. De nuevo recomendamos que su configuración se a multiplexada se use o no un arreglo.
Temporary Data Files	NO RAID, RAID0,	El principal requerimiento para estos archivos es un acceso rápido y no tanto la confiabilidad. Si el datafile se pierde, simplemente puede ser recreado.
Archived Redo Logs	NO RAID, RAID0, RAID 0+1	De nuevo recomendamos multiplexar en caso de no usar un arreglo. Si utilizamos una implementación RAID 0+1, dejemos que sea el hardware quien lo multiplexe.
Rollback/Undo Datafiles	RAID 0+1	Estos archivos tienen un acceso fuerte de I/O y deben estar protegidos. No pueden ser multiplexados por Oracle, por lo tanto dejemos que el hardware lo haga.
Datafiles	RAID 0+1, RAID5	La mayor parte del I/O a los datafiles son almacenadas en un buffer, con las escrituras realizándose en background. Como resultado, un RAID5 puede ser utilizado para reducir costos sin impactar tanto el desempeño. La recomendación es utilizar RAID 0+1 siempre que sea posible.

MS EXCHANGE SERVER		Tipo de arreglo y particionamiento
Número de discos		
2-4		RAID1. En dos particiones, una para el sistema operativo y la paginación. La otra para el Exchange Server RAID5 (Disco C). Tres discos para los archivos binarios y la base de datos
5		RAID1 (Disco D). Dos discos para los archivos de log. RAID5 (Disco C). Tres discos para los archivos binarios y la base de datos
6		RAID1 (Disco D). Dos discos para los archivos de log. NO RAID (Disco E). Un disco para la paginación.

Para el caso de SQL Server de Microsoft, la configuración es más simple. Se recomienda el uso de RAID1 y RAID5. RAID10 también es recomendado, pero dado el costo de su implementación, optamos por la primera opción. Con la configuración de los arreglos solventamos las necesidades de I/O de las bases de datos. Para bases de datos muy grandes se recomienda distribuirlas en múltiples arreglos de discos.

Los Transactional Log Files requieren de un acceso secuencial optimizado y deben de contar con un buen nivel de protección, por ello recomendamos RAID1. ©

Research In Motion BlackBerry 7100g

Research In Motion es el creador de la plataforma inalámbrica BlackBerry, que integra hardware, software y servicios soportados prácticamente por todos los estándares de redes inalámbricas. La diferencia de esta tecnología con otras similares, es que realmente ofrece una solución para los usuarios que necesitan mantenerse en contacto con su oficina a través de e-mail, mensajes de texto y multimedia, y por supuesto, telefonía celular, sin importar el lugar donde se encuentren. Dada la relación de RIM con proveedores de servicio y desarrolladores y fabricantes independientes, sus productos funcionan de una manera integral en cualquier red, sin necesidad de configuraciones adicionales, por lo tanto, los hombres de negocios que viajan constantemente, se sentirán como en casa con un dispositivo de éstos. El 7100g incorpora Bluetooth, pantalla LCD de alta resolución, tonos polifónicos y una batería de alta duración. Opera en bandas GSM y GPRS de 850/900/1800/1900 MHz en Norteamérica —México incluido—, Europa y Asia. Ideal para los viajeros que no pueden vivir sin e-mail y aplicaciones para Internet.



MGE – UPS Systems Galaxy 4000 UPS

Indispensable para toda empresa de TI, un UPS puede ser fundamental para resguardar la integridad de los equipos y la información contenida en ellos. Este modelo de MGE —empresa líder en el rubro—, incorpora tecnología que estabiliza el voltaje incluso en circunstancias extremas, como bajas repentinas y apagones prolongados. Su diseño lo hace de fácil instalación, y su amigable interfaz de usuario con pantalla táctil es sumamente sencilla de operar, incluso para quienes nunca han utilizado un UPS industrial.



Sony AIBO ERS-7M3

Este nuevo modelo del famoso can robótico de Sony es el primero en entender palabras en español —30 palabras y frases—, y tiene un gran diccionario de inglés. Además, goza de memoria a corto plazo para reconocer a las personas y objetos a su alrededor y, por si fuera poco, puede leer servicios de RSS desde Internet. La función para tomar fotos y mantener un diario sigue presente. Una mascota con más opciones y funcionalidad que un perro común, y también un dispositivo que impresiona a todos los asistentes a una junta o reunión.

Llevando una vida....

M G E
UPS SYSTEMS

Sin interrupciones



Av. Congreso de la Union 524
Santa Anita Iztacalco
México 08300
Tel. (52) 55389687 Fax (52)55307625
www.mgeups.com.mx

01

The Mythical Man-Month

Frederick B. Brooks Jr
Addison Wesley. Edición de 20 aniversario, 1995

The mythical man month es tal vez el libro más conocido sobre manejo de proyectos de software. Desgraciadamente, no podemos decir que es el que mayor influencia ha tenido, dado que 30 años después de su edición original, seguimos cometiendo los mismos errores por no hacer caso a esta obra.

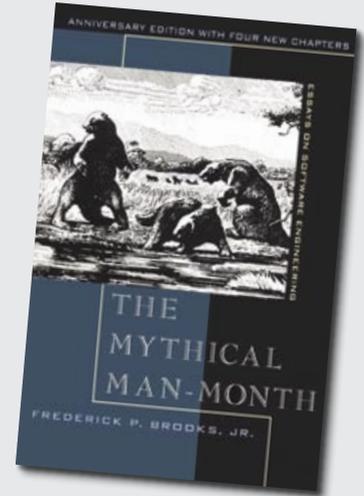
La edición original fue escrita en 1975, y la edición de 20 aniversario en 1995. Aunque las herramientas y tecnologías han cambiado desde entonces, la dinámica de los proyectos sigue siendo la misma. He aquí algunas de las perlas de sabiduría compartidas en esta obra:

El mítico mes-hombre. Las personas y los meses son intercambiables solamente cuando una tarea se puede dividir entre muchos trabajadores sin necesidad de comunicación entre ellos. Esto aplica a actividades como recoger algodón, pero no es ni remotamente aplicable al desarrollo de software. Cuando una tarea no se puede dividir por restricciones de secuencia, la aplicación de mayor esfuerzo no tiene ningún efecto en la duración. Gestar a un bebe toma nueve meses, sin importar cuantas mujeres se asignen a ello.

El efecto del segundo sistema. Cuando se diseña un sistema después de haber diseñado otro que fue pequeño y exitoso, habrá una tendencia a tratar de hacer un diseño grandioso en el nuevo sistema, y terminará convirtiéndose en una monstruosidad.

Peticiones de cambio y versiones del sistema. Ningún cliente o usuario sabe por completo lo que quiere en el sistema a construirse. Conforme el sistema comienza a tomar vida, y el cliente interactúa con él, entiende lo que realmente quiere del sistema y, por consecuencia, solicita cambios. Estos cambios deben incorporarse, pero sólo hasta cierto límite, después del cual los requerimientos se deben congelar. Todas las peticiones de cambio se deberán posponer hasta la siguiente versión del sistema o, de otra manera, éste nunca se terminará.

Si de alguna manera estás involucrado con proyectos de software, debes leer este libro, pero más que nada, debes hacerle caso.



02

Project Retrospectives: A Handbook for Team Reviews

Norman L. Kerth
Dorset House Publishing, 2001

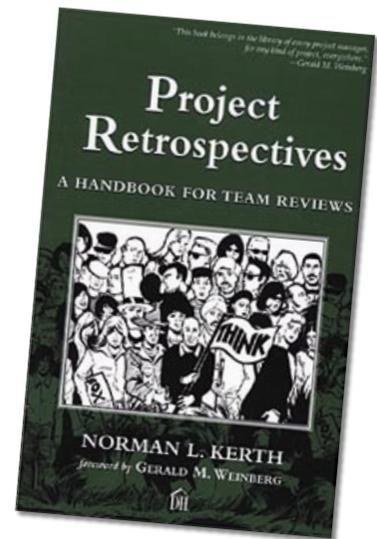
En su columna de este número, Luis Cuellar nos habló sobre las sesiones de lecciones aprendidas al final de un proyecto. Precisamente, este libro es una guía para sacar el máximo provecho posible de estas sesiones, denominadas por el autor como “retrospectiva del proyecto”. El autor trata las retrospectivas como un ritual, y a lo largo del libro comparte técnicas y herramientas que debe aplicar el facilitador de este ritual.

El ritual típicamente requiere tres días de sesiones grupales, que de preferencia se realizan fuera de la oficina. Reconociendo que será difícil obtener la aprobación y presupuesto por parte de la Dirección, el autor incluye tips sobre cómo convencerlos, y hasta un “checklist” para generar una propuesta. Previo a las sesiones, el facilitador obtiene

retroalimentación sobre los principales puntos a cubrir, y pide a los involucrados que junten documentos del proyecto, tales como planes, minutas, bitácoras de pruebas, etc.

Algo en lo que el autor pone gran énfasis, es en la seguridad. Esto es, que los involucrados no se sientan amenazados, ni en peligro por lo que suceda en estas sesiones. Una retrospectiva no se trata de encontrar culpables. Se asume que todos hicieron el mejor trabajo posible con el tiempo e información con que contaban. El propósito del facilitador es lograr que los miembros compartan sus observaciones sobre el proyecto, de manera que los diferentes miembros no se sientan atacados. El facilitador no debe mostrar su propia opinión, simplemente ayudar a otros a expresar la suya.

Como sabemos, gran parte del éxito de un proyecto de software depende de la comunicación interpersonal, y las retrospectivas pueden ayudar a un equipo a mejorar su comunicación para tener éxito en los proyectos futuros.



Recomienda un libro para esta sección, escribe a:
biblioteca@softwareguru.com.mx

Desde 1997 la **amcis** ha reunido a los profesionales y académicos interesados en promover la calidad en la Ingeniería de Software.

Nuestro objetivo es compartir el conocimiento y experiencia a través de:

- ✓ Diplomado en Calidad de Software (único en México)
- ✓ Conferencias y reuniones mensuales
- ✓ Seminarios



Si quieres aprender lo mejor de la Calidad en Ingeniería de Software y compartir tus experiencias **ven a conocernos.**



Mayores informes: www.amcis.org.mx,
 e-mail: info_amcis@yahoo.com.mx, Tel.: 5563.3059
 Entrada gratis a las conferencias y reuniones mensuales presentando este anuncio.

■ INDEX

DIRECTORIO

TENEMOS UN ESPACIO RESERVADO PARA TI

Si deseas anunciarte contáctanos en el (55) 5239 5502 o en ventas@softwareguru.com.mx

<i>Anunciante</i>	<i>Páginas</i>	<i>Sitio</i>
AMCIS	55	www.amcis.org.mx
Avantare	19	www.avantare.com
e-Quality	07	www.e-quallity.net
EduTECSA	39	www.edutecsa.com
Gopac	45	www.gopac.com.mx
Grupo STI	51	www.gsti.com.mx
IBM	F4	www.ibm.com/mx
Imexsoft	31	www.imexsoft.com.mx
Innevo	35	www.innevo.com
Itera	33	www.itera.com.mx
LinuxWorld	F3	www.linuxworldexpo.com.mx
MGE	53	www.mgeups.com
Microsoft	F2-1	www.microsoft.com/mexico
Milestone	09	www.milestone.com.mx
Roca Sistemas	37	www.rocasistemas.com.mx
SafeNet	11	www.la.safenet-inc.com
Ssistemas	41	www.ssistemas.com
Tcds	47	www.tcds.com.mx
Vision Consulting	35	www.visionconsulting.com.mx

Ser, Hacer, Crecer

BASES PARA EL CRECIMIENTO SUSTENTABLE

Por Ernesto Corona

Watts Humphrey dice en su libro “Introducción al PSP^(SM)”, “...nada te dará más satisfacción, pública y privada, que realizar trabajo con calidad. Te dará seguridad, confianza, y en corto o largo plazo, te será reconocido como un gran trabajo...”.

La reflexión es: ¿Hacer o Ser? Observemos nuestro comportamiento ante algunas situaciones comunes.

Ser

Ser está asociado a lo momentáneo, a un lugar que se ocupa, o algo que se obtiene. Acostumbramos asociar el ser con tener. Soy lo que soy por tener lo que tengo. Primero poseo, luego soy. Hay ocasiones en que ser se transforma deliberadamente en dejar de ser. Y esto causa dolor, pérdida de confianza. Ya no somos (los primeros, los líderes, los distinguidos, los de moda). Entonces somos la ausencia de aquella cualidad distinguida, nos falta una “parte acostumbrada”, y duele.

Hacer

Cuando la satisfacción viene de hacer, hay algo que nos permite trascender y multiplicarnos: allí está lo que hicimos, y su calidad es nuestro sello, orgullo, satisfacción.

“Caminante no hay camino, se hace camino al andar”. “Ser camino” está determinado por la acción del caminante y su decisión de andar, descubrir, fundar, disfrutar. Podemos hacer mucho más que ser. Y podemos mejorar cotidianamente la manera en que hacemos.

Crecer

Hay quienes confunden el crecer con engordar, pero hay una gran diferencia. Engordar nos provoca ocupar nuevos espacios, a veces hasta de manera indeseada. Perdemos agilidad y descubrimos que tenemos que realizar mayor esfuerzo para obtener lo mismo de antes. Nos cansamos.

¿Cómo se relaciona esto con nuestras organizaciones? Las empresas engordan al alimentarse mal, y en exceso, de proyectos que no son para ellas. De hecho, puede engordar una organización sin moverse, por proyectos que le caen encima. Proyectos chatarra. Una organización crece cuando hace ejercicio para crecer: me alimento de lo que me hace falta, me educó en lo que necesito, y sé claramente cómo me quiero ver en el tiempo. Aprovechar oportunidades no es comer basura. Decir que sí a todo, lo es. Si recomendamos a las personas no tener ese comportamiento, ¿por qué lo aceptamos en nuestras organi-

zaciones? No todas las empresas pueden hacer todo tipo de proyectos. Más aún: las recién nacidas deben alimentarse cuidadosamente (seleccionar proyectos), y procurar que ellos tengan los mejores padres posibles (equipos).

Puede que por momentos aumenten las ventas y nos sintamos en pleno crecimiento. Aumentar las ventas no necesariamente es crecer. Puede, inclusive, ni siquiera aumentar nuestra utilidad. Esto es común en empresas de servicios, donde la entrega demanda un proceso más complejo que en compañías orientadas a productos. Si repentinamente nos encontramos con un aumento de las ventas y tras esto no existe infraestructura de conocimiento, respaldo por el producto, factores de diferenciación soportados y aprendizaje asociado a cada momento, no esperemos crecimiento asociado a las altas ventas. Simplemente demos gracias al contexto favorable.

¿Por qué es mejor que una empresa crezca a diferencia de engordar? Porque para crecer, necesariamente hay que hacer. Esto significa involucrarse, preocuparse, generar, participar. Dejar rastro. Hacer camino. Si nuestra estrategia como empresa de software se centra en el conocimiento, más vale que nos ocupemos en generarlo y mantenerlo. Y si hemos sido exitosos, sepamos cómo repetirlo. Y si ya logramos repetirlo, y es parte de nuestra cultura aprender de nosotros y de otros, ser críticos, abiertos y buenos administradores, ya tenemos la cultura necesaria para que el cambio de nivel “Inicial” a nivel “Gestionado” de CMMI duela casi nada.

Si afortunadamente vendemos más por obra y gracia de factores externos, agudicemos el ingenio para transformar esto en crecimiento sustentable. En general, engordar agudiza los problemas

de comunicación, administración, y puede llevarnos a dejar de vender, o vender mal. Estamos lentos, nos sentimos grandes (no fuertes) y hasta nos da miedo seguir adelante.

Es claro que hacer implica riesgos y genera miedos. Implica no poder hacer todo, enfocarnos en lo que podemos y se relaciona más cercanamente con el plan de crecimiento, mirando los objetivos estratégicos de la organización.

Sin dudas, hacer es algo que ocurre a todos los niveles en las empresas. Con pocos o muchos recursos. Adquirir conocimiento y transformarlo en capital intelectual de la organización es hacer para crecer. Y cuando tengamos la oportunidad de hacer, exijárnos la mayor calidad posible. Medirnos, cambiarnos, mejorarnos y seguir haciendo. Mejor para mí, mejor para el entorno, mejor para todos.

En este juego saludable de hacer, y retomando la frase de Watts Humphrey inicial, la orientación debe ser “hacer con calidad”. Y la calidad está condicionada por los recursos, tanto como el hacer está condicionado por los mismos. Pues entonces tengamos inventiva para generarlos, comenzando desde lo mínimo inmediato que necesitamos para dar el primer paso. Incorporar disciplinas de gestión es hacernos de recursos. Poner en práctica PSP o patrones de trabajo como PDCA (*Plan, Do, Check, Act*) es beneficioso a corto plazo. Seleccionar modelos de mejora como CMMI o MoProSoft, comprenderlos, cuestionarlos y aplicarlos, es generamos recursos.

De todo esto se compone el hacer: Involucramiento, mejora, diferenciación y capital intelectual. Las bases del crecimiento sustentable en nuestras compañías. 

Ernesto Corona es Vicepresidente de Innevo, empresa mexicana dedicada a la mejora de procesos de software, desempeñando el cargo de Director de Calidad de Productos y Servicios. Es además socio consultor para Vates y Kanav, ambas empresas argentinas acreditadas CMM 3 y CMMI 2 respectivamente. ecorona@innevo.com

¡Por primera vez en México!



Febrero 14-17, 2006



LinuxWorld Conference & Expo México 2006 será la más grande reunión para conocer las novedades de empresas tanto nacionales como internacionales, sus ofertas, productos, aplicaciones y soluciones dirigidas a profesionales de Linux y del Open Source en México en tan sólo 4 días.

Asista al programa de conferencias más importante de la industria

Miércoles 15 de febrero, 2006

TOP TECHNOLOGY	LINUX EMPRESARIAL	GOBIERNO
Cómo el open source ha cambiado la tecnología de información para siempre	El rol dinámico que Linux juega en una empresa	Avances y retos de Linux y OSS en la Administración Pública y Finanzas
Middleware; Servers Applications (Apache, JBoss, Websphere, Zope, Zend)	El valor de retorno sobre inversión (ROI) de usar Linux	Historias de éxito en México
MySQL, las mejores prácticas de clustering, respaldo y recuperación	El Top Ten de OSS para Directores de Informática	Cómo mezclar ambientes en entidades gubernamentales
Oportunidades de negocio en América Latina para el mercado de Linux	El arte y la ciencia de evitar problemas de seguridad	Oportunidades de negocio y perspectivas en el gobierno para Linux

CONFERENCIA MAGISTRAL Efrain Rovira, director mundial de Linux Marketing, HP

Jueves 16 de febrero, 2006

CONVERGENCIA	TOP TECHNOLOGY	PyMES
Integrando Linux a la infraestructura de una organización IT	El Open Source PBX por Asterisk, Digium	Ventajas técnicas y económicas de implementar Linux en PyMES
Arquitectura de red para voz, video y datos	Extensión de la movilidad empresarial; en cualquier lugar y en cualquier momento	Buenas prácticas en ambientes Linux
Samba, la interoperabilidad entre Windows y Linux	El futuro de las comunicaciones y las tecnologías libres	Linux desktop para PyMES
Migración de Windows a Linux	El poder de la tecnología de la siguiente generación	5 sugerencias para desarrollar un paquete exitoso de soluciones Linux para PyMES

CONFERENCIA MAGISTRAL Miguel de Icaza, CTO Ximian, Novell

Viernes 17 de febrero, 2006

TUTORIALES		
Haciendo de Linux una opción segura	Tips & tricks para Apache	Administración de Sistemas Linux
Soluciones para macrocentros de datos	Herramientas para soluciones OSS gubernamentales	Opciones técnicas para respaldo y recuperación de información

CONFERENCIA MAGISTRAL

Pre-regístrese a la exposición sin costo...

www.linuxworldexpo.com.mx

Organizado por:



E.J. KRAUSE DE MÉXICO



IDG WORLD EXPO

Mayores informes sobre costos y conferencias:
Martha Medina • Coordinadora de Proyectos Especiales
Tel. (52-55) 1087-1650 ext. 1139 mmedina@ejkrause.com

**Cupo Limitado
en el Programa de
Conferencias**



USTED VS. LAS COMPLICACIONES PARA INTEGRAR APLICACIONES



WebSphere

La solución de Middleware IBM WebSphere Software es la forma más fácil de integrar una multitud de aplicaciones. **De hecho, usted puede conectar cualquier aplicación para cualquier plataforma con IBM Middleware de estándares abiertos.** Sólo IBM cuenta con años de experiencia probada ayudando a sus clientes a construir aplicaciones compuestas, por eso la solución de estándares abiertos para aplicaciones complejas, plataformas y combinaciones en infraestructura de TI es IBM WebSphere que le permite reutilizar sus activos actuales de TI. Imagínese todo lo que puede ganar aumentando su eficiencia para hacer su negocio más flexible.

DESCUBRA CÓMO EN: IBM.COM/MIDDLEWARE/CONNECT