

ISSN:2007-5448



2nd INTERNATIONAL
CONFERENCE ON
SOFTWARE
PROCESS
IMPROVEMENT



Número Especial

Editado por:

Dra. Mirna Ariadna Muñoz Mata

Dr. Jezreel Mejia Miranda



RECIBE

Revista electrónica
DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICAS Y ELECTRÓNICA

Vol. 2 No.3

ReCIBE Revista electrónica de Computación, Informática, Biomédica y Electrónica, Año 2 No. 3 Noviembre - Diciembre 2013, es una publicación semestral editada por la Universidad de Guadalajara, a través de la División de Electrónica y Computación del CUCEI. Av. Revolución #1500 Modulo "O" Col. Olímpica, C.P. 44830, Guadalajara, Jalisco, México. Teléfono: +52 (33) 1378-5900, <http://recibe.cucei.udg.mx>, recibe@cucei.udg.mx. Editora responsable: Adriana Peña Pérez Negrón. Reserva de Derechos al Uso Exclusivo No. 04-2012-091818233000-203, ISSN: 2007-5448, otorgados por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número: David Bonilla Carranza. Fecha de la última modificación 9 de Diciembre del 2013, con un tiraje de un ejemplar. Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación. Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización de la Universidad de Guadalajara.

Índice

Editorial

Presentación.....i

Computación e Informática

Adquisición de Tecnología: Un Modelo de Gestión.....I

Rafael Rincón Bermúdez, Guadalupe Peláez Ramírez.

Herramienta para la Automatización de la Revisión Sistemática.....II

Edgar Felipe Uribe Rayas, Jorge Alexis Márquez del Real, J.
Guadalupe Uribe Dévora, Claudia Valtierra Alvarado, Jezreel
Mejia Miranda.

Identificando Hallazgos de Mejora en Pymes de TI
Utilizando un Modelo Ontológico para CMMI-DEV v1.3.....III

J. Guadalupe Uribe Dévora, Edgar Felipe Uribe Rayas,
Jorge Alexis Márquez del Real, Claudia Valtierra
Alvarado.

Marcadores para la realidad aumentada para fines educativos.....IV

María Reina Zarate Nava, Cecilio Francisco Mendoza González,
Honorato Aguilar Galicia , Juan Manuel Padilla Flores.

Metodologías para desarrollar software seguro.....V

Carlos Joaquín Brito Abundis.

Herramienta de Gestión de Modelos Ontológicos Aplicados
a la Mejora de Procesos Software.....VI
José Eduardo Guadalupe Gaytán Solís, Ricardo González Saldívar,
Edrisi Muñoz Mata.

Selección de Estrategias para la Implementación de Mejoras
de Procesos Software.....VII
Brenda Liliana Durón del Villar, Mirna Ariadna Muñoz Mata.



Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Presentación

Nos es muy grato participar en la edición de este número especial de la Revista RECIBE dedicado a la temática de Ingeniería de Software (IS). Los artículos presentados en este número fueron seleccionados de entre los aceptados para el segundo Congreso Internacional de Mejora de Procesos Software CIMPS 2013, realizado en la Ciudad de Zacatecas, México del 2 al 4 de Octubre del 2013. Cabe resaltar que la tasa de aceptación del CIMPS ha sido del 30% de los artículos recibidos.

En los últimos años el software se ha convertido en el núcleo principal en la mayoría de las empresas ya que la capacidad de las organizaciones y sus productos, sistemas y servicios que les permite competir, adaptarse y sobrevivir, depende cada vez más del software e incluso es mediante él por el que las empresas pueden adaptarse de manera rápida y fácil a diferentes sectores del mercado. Es por ello, que el software juega un papel importante dentro de cualquier organización tanto en el ámbito industrial como en el académico.

Dentro de este número especial se presentan investigaciones que cubren tanto el entorno académico como el industrial. Los artículos dirigidos al entorno académico se enfocan en presentar por un lado un proceso para la adquisición de nueva tecnología en las universidades; y por otro lado, las herramientas que soportan y fortalecen la realización de investigaciones mediante métodos formales como lo es la revisión sistemática y la enseñanza de materias específicas como es el caso del uso de realidad aumentada para material de laboratorio de la materia de Química.

Los artículos sobre el entorno industrial se enfocan en proporcionar a la industria herramientas para la seguridad y mejora de procesos. En el campo de la seguridad se presenta un análisis de metodologías para el desarrollo de software seguro y en el campo de la mejora de procesos se abordan tres aspectos clave para apoyar a las organizaciones en la mejora de sus procesos: un artículo sobre una herramienta que permite la gestión del conocimiento mediante el uso de modelos ontológicos basado en SWEBOK; también una experiencia en la identificación de hallazgos de mejora en PyMES de TI utilizando un modelo ontológico para CMMI-DEV v1.3; y finalmente se presenta un modelo que permite la selección de estrategias para la implementación de mejoras.

Esperando que este número especial sea de utilidad en el campo de la Ingeniería del software, agradecemos todas sus contribuciones y damos la enhorabuena a todos aquellos autores cuyos artículos han sido publicados en este número especial de RECIBE.

Dr. Mirna Ariadna Muñoz Mata

Dr. Jezreel Mejia Miranda

**Centro de Investigación en Matemáticas- Unidad
Zacatecas**

Adquisición de Tecnología: Un Modelo de Gestión

Rafael Rincón Bermúdez

**Universidad EAFIT
Medellín, Colombia
rdrinconb@gmail.com**

Guadalupe Peláez Ramírez

**Politécnico Colombiano Jaime Isaza Cadavid
Medellín, Colombia
guapera802@gmail.com**

Resumen: En la actualidad, las empresas basan su gestión en la utilización de la información, por lo cual necesitan el apoyo de los recursos tecnológicos para facilitar esta labor, sin importar la razón social o el sector productivo al que pertenezcan. Es por ello, que las Tecnologías de la Información juegan un papel muy importante para la toma de decisiones, brindando en el menor tiempo posible información organizada. Sin embargo, en muchas ocasiones por responder a necesidades internas o externas de manera precipitada, se adquieren tecnologías de información sin un adecuado proceso de análisis organizativo y establecimiento de requerimientos, lo que implica en algunos casos, una errada selección de los proveedores o una adquisición de tecnología equivocada. Este modelo se fundamenta en unos referentes normativos tanto internos como externos, así como en unos componentes teóricos, asociados a cada una de las fases que lleva éste y que servirán para la comprensión del mismo. Además, el Modelo de Gestión para la Adquisición de Tecnologías de Información, servirá como base para llevar a cabo de

forma organizada el proceso a seguir en la contratación de un servicio o producto en materia de Tecnologías de la Información.

Palabras clave: Adquisición de tecnología, tercerización, modelo de gestión, calidad, gestión de proyectos.

Technology Acquisition: A Management Model

Abstract: Currently, the companies base their management on the use of the information, so that's why they need the support of technological resources to facilitate this work, regardless of the company name or the productive sector from which they belong. Therefore, information technology plays an important role in decision-making, providing information organized in the shortest time as possible. However, on many occasions to respond to internal or external needs precipitously, information technologies are acquired without an adequate organizational process of analysis and establishment of requirements, which in some cases means a wrong selection of suppliers or acquisition of wrong technologies. This model is based on internal and external policies as well as theoretical components associated with each of its phases, which will serve as an understanding of the model itself.

Keywords: Acquisition of technologies, outsourcing, management model, quality, suppliers, project management.

1. Introducción

La adquisición de tecnología cada vez cobra más importancia para las empresas de cualquier sector productivo u objeto social, pues la variable tecnológica, se ha convertido en elemento estratégico para su desarrollo y soporte fundamental en el desarrollo de sus operaciones y globalización, al facilitar la interacción y comunicación de los mercados de todo el mundo.

Sin embargo, en muchas ocasiones, por responder a necesidades internas o externas de manera precipitada, se adquieren tecnologías de información sin un adecuado proceso de definición y análisis de requerimientos, lo que implica en algunos casos selección de proveedores y adquisición de tecnología que no está relacionado efectivamente con el desarrollo del negocio.

El Modelo de Gestión de Adquisición de Tecnologías de la Información (MOGATI), que se propone, plantea llevar a cabo de forma organizada el proceso a seguir desde la definición hasta la contratación de un servicio o producto de TI, integra en un proceso claramente definido los aspectos legales, los procesos internos de la organización y la calidad, tal que integrados propongan actividades, roles y productos que aseguren una selección de tecnología acorde con los requerimientos.

El Modelo se validó en el Politécnico Colombiano Jaime Isaza Cadavid, Institución Universitaria de carácter público, igualmente, se sometió a juicios de expertos de entidades públicas y privadas con el fin de verificarlo y ajustarlo.

En la sección 2 se describen los antecedentes del modelo propuesto; en la sección 3 se presenta el marco de referencia del modelo; en la sección 4 se describe el modelo propuesto y en la sección 5 se muestran las conclusiones y trabajos futuros.

2. Antecedentes

Las organizaciones y empresas de consultoría tienen definidos procesos de adquisición de TI, que en alguna medida no integran todos los elementos de un proceso de esta naturaleza. Para el presente trabajo, se analizaron los modelos de gestión descritos por Amador y Márquez (2009), el modelo de gestión de una entidad financiera local, el modelo de adquisición de TIC de “la Caixa”, etapas del modelo de gestión estratégica de tecnología de Pedroza (2001), el modelo de gestión del Plan Estratégico de Tecnologías de la Información de Azurian (2012); todos estos proponen una secuencia lógica para la adquisición de TI, pero cada uno ajusta ese proceso a la arquitectura empresarial que han definido y a sus flujos internos de trabajo, convirtiendo a cada modelo en único, dejando a la organización como responsable de la ejecución o no de algunas actividades.

También se determinó como factor clave de la adquisición de TI, el aumento a nivel mundial de las inversiones en tecnología tanto en países desarrollados como en desarrollo, teniendo en cuenta que estos últimos están mejorando su acceso a los diferentes recursos, como la movilidad, la seguridad, los servicios digitales, el almacenamiento de información, entre otros. Cifras como las de la consultora IDC (International Data Corporation) en su informe Latin America Predictions del mes de febrero de 2013, prevé que Latinoamérica, será la región con el crecimiento más rápido de TI en el mundo, teniendo en cuenta que el hardware equivaldrá al 66% del crecimiento, el software el 13% y los servicios el 21% restante, y que México, Brasil y Colombia serán los mercados de mayor crecimiento, y por ende, los que más adquieran tecnologías de la información. Sin embargo, en la actualidad se presentan obstáculos que deben ser tenidos en cuenta a la hora de invertir y adquirir tecnología, como déficit en infraestructuras, inestabilidad de las reglas institucionales, derechos de propiedad mal definidos, innovación tecnológica, entre otros. Por tal motivo, el fortalecimiento de las capacidades para organizar y estandarizar los procesos de identificación de necesidades tecnológicas se hace fundamental para su adquisición.

3. Marco de Referencia

En este trabajo, la gestión de tecnología se considera integrada por dos componentes principales: La Gestión Estratégica de TI y la Gestión Operativa de TI.

Gestión Estratégica de TI: Los elementos estratégicos hacen referencia a las políticas y objetivos que direccionan a la institución y definen las estrategias a seguir para alcanzar los resultados esperados.

Las tecnologías de la información se han convertido en una parte dominante para la realización de los procesos del negocio o misionales, y las instituciones de educación superior no son ajenas a este fenómeno, ya que no sólo tienen

un enfoque operacional sino también estratégico, trayendo nuevos retos para los líderes institucionales, afectando no sólo las áreas administrativas sino también las académicas. La negociación de tecnología y su adquisición, ocurre cada vez con más frecuencia, y se ha vuelto más compleja y más crítica para el éxito del desarrollo de la organización, involucrando mayor número de personal con habilidades necesarias para negociar de forma satisfactoria.

La tecnología debe implementarse ordenadamente siguiendo los parámetros creados por la planeación estratégica en la Institución, identificando los recursos tecnológicos más importantes que agregarán valor y mejorarán la prestación del servicio, haciendo énfasis en permitir la aprehensión de nuevos conocimientos en todos los niveles del Politécnico. Este componente debe iniciar desde la estrategia institucional, plasmada en planes y políticas, principalmente relacionadas con las tecnologías de la información, y las necesidades macro a nivel tecnológico deben ser identificadas desde la definición de las metas y objetivos institucionales, que permitan su adecuado desarrollo.

A nivel institucional, este componente debe tener en cuenta, en las políticas para la Gestión del Proceso de Tecnología de la Información, lo siguiente:

- Los objetivos institucionales deben estar definidos y acordados antes de iniciar la adquisición de cualquier solución de TI.
- La adquisición e implementación de tecnología en la Institución debe ser un proceso planeado que responda a las necesidades y objetivos generales y soporte los procesos clave identificados. De otra forma, los costos de introducción, implementación e interoperabilidad pueden crecer de manera insostenible, al tiempo que el valor agregado de su uso se reduce significativamente. La arquitectura empresarial y tecnológica es el primer paso para poner a la tecnología en el lugar que más aporte, brindando soluciones a las unidades de gestión y procesos misionales.

- Construir estratégicamente la gestión tecnológica a partir de directrices institucionales.
- Definir intenciones estratégicas en tecnología, partiendo de la arquitectura institucional.
- Identificar habilidades tecnológicas en todo el personal (actuales y necesarias).
- Determinar actores tecnológicos que puedan ser partícipes constantes en procesos de adquisición de tecnología, y que cuenten con las habilidades para ello.
- Explorar ambientes tecnológicos que puedan aportar valor a la Institución, tanto administrativa como académicamente.
- Pronosticar, analizar y auditar la tecnología actual como punto de partida para las mejoras futuras.
- Construir mapas tecnológicos para la planeación de nuevas adquisiciones y para la implementación de nuevas tecnologías, soportados en la arquitectura tecnológica existente.

Gestión Operativa de Adquisición de TI: Este componente propone un proceso para la adecuada adquisición de tecnología, que inicia con la identificación de necesidades desde cada unidad de gestión institucional, pasando por el reconocimiento de tecnologías que aporten soluciones a las necesidades identificadas, luego seleccionándolas, negociándolas, implementándolas, adaptándolas, utilizándolas y asimilándolas y, terminando en la transferencia de conocimiento y lecciones aprendidas, con el fin de mejorar los procesos definidos. Todo lo anterior, descrito en fases que contienen actividades, responsables y entregables que deberán estar alineados con la legislación externa, y normatividad interna que gobierna al Politécnico Colombiano Jaime Isaza Cadavid, por su naturaleza de Institución Universitaria de carácter público.

El modelo propuesto integra componentes teóricos como base para su fundamentación y como punto de partida para tener que integrar todos los

elementos conceptuales en un esquema organizado, como lo muestra la Figura 1.

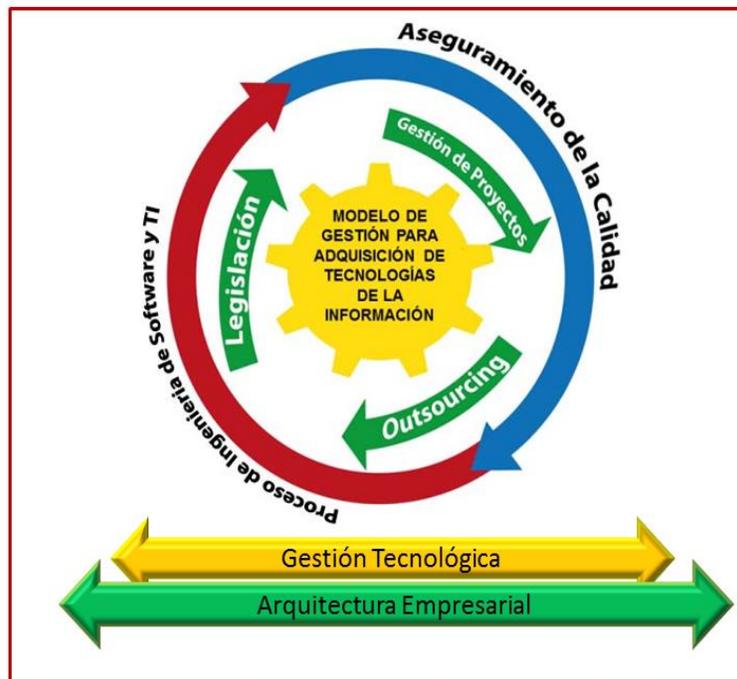


Figura 1. Componentes teóricos Modelo de Gestión para la Adquisición de Tecnologías de la Información.

En la figura anterior, la arquitectura empresarial relaciona el objeto institucional con las tecnologías de la información; la gestión de proyectos hace referencia a todas las actividades que se realizan para cumplir con un fin principal definido, en un tiempo establecido utilizando recursos tanto humanos como materiales y para el cual se deben tener presupuestados los costos en que se incurrirán; la gestión tecnológica es una práctica soportada en un conocimiento derivado del análisis y la interpretación de las observaciones del comportamiento del desarrollo tecnológico; el outsourcing es la transferencia de un proceso o una fase del negocio a un tercero especializado con el objetivo de agilizar y mejorar los procesos, tiene relación con este modelo dado que es frecuente que la variable TI sea tercerizada por las empresas en la actualidad. El Modelo debe tener como componente transversal el concepto de calidad, entendido como el nivel de cumplimiento de necesidades o expectativas establecidas por los

clientes, a partir de unos rasgos o elementos diferenciadores de la organización.

4. Modelo de Gestión para la Adquisición de Tecnologías de la Información (MOGATI)

Las fases propuestas por el modelo mostrado en la Figura 2 son cuatro, y tienen un orden de ejecución, desde la primera y siguiendo continuamente hasta la cuarta, donde serán recogidas las lecciones aprendidas que retroalimentarán la Gestión Estratégica de TI, a partir de mejoras a tener en cuenta para futuras adquisiciones de tecnología.

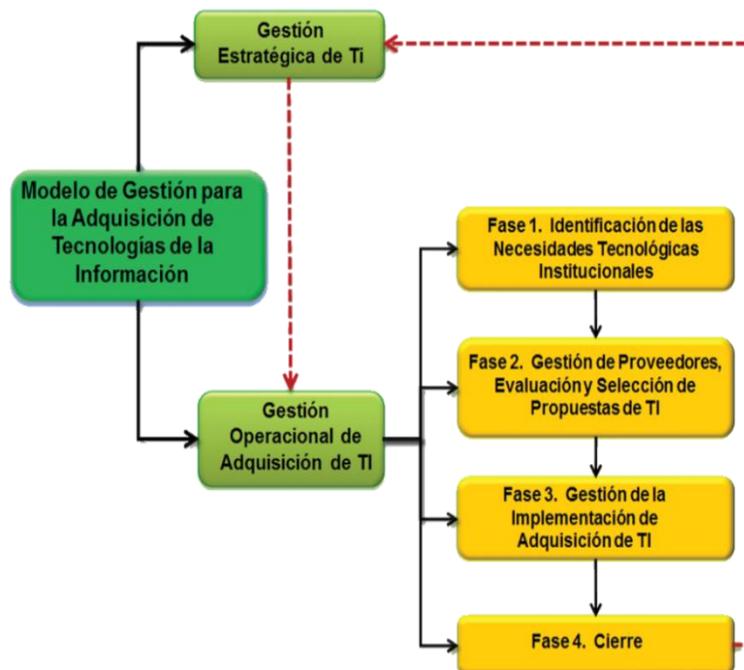


Figura 2. Modelo de Gestión para la Adquisición de Tecnologías de la Información para el Politécnico Colombiano Jaime Isaza Cadavid.

Fase 1. Identificación de las necesidades tecnológicas institucionales.

Esta es la primera fase del Modelo de Gestión para la Adquisición de Tecnologías de la Información, y en ella se propone que cada unidad de gestión al identificar una necesidad de mejora, analice si ésta se puede solucionar a partir de la adquisición de tecnología o mejorando la existente, realizando un análisis que contenga los siguientes aspectos: Qué, por qué, para qué, cuándo, dónde y cómo, reflejados en la definición de requerimientos. Con la definición de requerimientos se expresa la necesidad que tiene la unidad de gestión o la Institución para adquirir una tecnología, donde se especifique de forma clara y precisa los aspectos más relevantes para la adquisición. Para realizar tal definición es necesario tener muy claras las necesidades que originan el requerimiento a nivel Institucional.

Es importante tener conocimiento de las empresas que proveen el producto u ofrecen los servicios que se necesitan, con el fin de no sesgar el estudio por no contar con suficiente información de peso para establecer el presupuesto.

Esta fase aplica desde la definición de las necesidades tecnológicas hasta la identificación de las tecnologías de la información potenciales y que ofrecerán soluciones a estas necesidades.

Fase 2. Gestión de Proveedores y Evaluación de Propuestas de TI.

La segunda fase del Modelo de Gestión para la Adquisición de Tecnologías de la Información propone que se realice un proceso claro de evaluación de propuestas y selección de proveedores, teniendo como punto de partida la descripción detallada de los requerimientos especificados en la Fase 1, y elaborando a partir de ellos, una solicitud de propuestas o RFP (en inglés: Request For Proposal) que contendrá las bases para que los oferentes o

proveedores potenciales realicen sus propuestas técnicas, funcionales, de proyecto y económicas, de la tecnología que se requiere adquirir, todo enmarcado bajo la naturaleza jurídica de la organización, y obedeciendo a las directrices definidas en los manuales internos de la compañía si los hay.

La selección del proveedor de la solución de TI está compuesta por el análisis formal de las propuestas de los proveedores y el análisis de las diferentes modalidades de adquisición. Esta selección debe responder a los planes, proyectos y programas que desarrollan las diferentes áreas de la Institución, en relación con la misión y la visión, ajustados a los Planes de Desarrollo, la programación presupuestal, así como al efectivo y permanente mejoramiento de la calidad de los procesos.

Esta fase aplica desde el refinamiento de los requerimientos definidos en la fase 1, hasta la selección del proveedor o proveedores de la tecnología a adquirir que dará solución a la necesidad planteada por la Unidad de Gestión involucrada.

Fase 3. Gestión de la Implementación de Adquisición de TI.

Para la tercera fase del Modelo propuesto, se describen unas actividades de manera general, sin entrar en detalles, ya que la fase de implementación variará, de acuerdo con la tecnología adquirida.

Esta fase inicia con la planificación de la implementación de la tecnología que se adquirió, y para esto debe existir una comunicación constante y clara entre el supervisor asignado al contrato y el proveedor seleccionado, y culmina con la implementación de la tecnología.

Para el buen funcionamiento del proyecto en esta fase, es necesario mantener un registro de la información comunicada y las decisiones tomadas. Asimismo, todos los documentos relativos al proyecto y sus versiones, contratos, todos

estos documentos almacenados y publicados, de manera que pueden ser consultados por cualquier miembro del equipo.

Hay que tener en cuenta que durante el ciclo de vida del contrato pueden haber cambios y reclamaciones, así como nuevas oportunidades de desarrollo. El principal objetivo tras la concesión de un contrato es el control del riesgo. Se debe mantener en todo momento el control sobre el proyecto y asegurarse que se cumplen los requisitos de tiempo y costos.

Como resultado de esta fase se elabora un documento detallado de la arquitectura e instalación de la solución, componentes, soluciones a problemas que se puedan presentar, futuras actualizaciones, información de contactos para mantenimiento y mejoras, entre otros.

Esta fase aplica desde el inicio de toda la gestión para la implementación de la tecnología adquirida hasta su instalación y puesta a punto.

Fase 4. Cierre.

Esta es la última fase del Modelo y en ella es necesario definir las actividades relacionadas con la finalización del contrato e interacciones requeridas para establecer y cerrar todo el acuerdo contractual establecido para el proyecto, así como aquellas que respaldan el cierre administrativo. Esta fase implica tanto la verificación final de la tecnología implementada (todo el trabajo completado de forma correcta y satisfactoria) y la actualización de todos los registros del contrato para reflejar los resultados finales para su uso futuro, además, el análisis del proceso para gestión del conocimiento e identificación de acciones de mejora. Algunos aspectos que deben ser tenidos en cuenta en esta fase, son:

- Verificar el cumplimiento del contrato (desde su alcance).
- Registrar adecuadamente la obtención de informes, autorizaciones y demás documentos derivados.

- Resolución de diferencias.
- Aseguramiento del cumplimiento de las obligaciones posteriores.
- Registro de toda la documentación, así como lecciones aprendidas.

Antes de la finalización del contrato, se debe revisar el mismo para encontrar aquellos aspectos en los cuales alguna de las partes debe mantener unas responsabilidades posteriores. Algunas de ellas pueden referirse a:

- Protección de la propiedad intelectual.
- Licencias.
- Servicios de mantenimiento, soporte y garantía.
- Actualizaciones de producto, nuevas versiones.
- Conservación de la información y los registros.

En esta fase de cierre, es fundamental que se realice el despliegue formal de la tecnología adquirida, mediante una reunión de inicio en producción de la solución comprada e implementada, con todos los involucrados en la misma. Esta fase aplica desde la planeación de la capacitación y entrenamiento del personal hasta la recolección y transferencia de lecciones aprendidas, que servirán como un componente de mejoramiento para la Gestión Estratégica de TI.

5. Caso Politécnico Colombiano Jaime Isaza Cadavid

El Politécnico Colombiano Jaime Isaza Cadavid como Institución Universitaria de carácter público, tiene las particularidades definidas en la Tabla 1.

LEGAL	
Ley 80 de 1993. Por la cual se expide el Estatuto General de Contratación de la Administración Pública.	Ley 1150 de 2007. Por medio de la cual se introducen medidas para la eficiencia y la transparencia en la ley 80 de 1993 y se dictan otras disposiciones generales sobre la contratación de recursos públicos.
NORMATIVIDAD INSTITUCIONAL Y OTRAS	
Políticas de Informática Corporativa. Acuerdo 09 de septiembre de 2010	
Manual de Contratación Politécnico Colombiano Jaime Isaza Cadavid – MLH01.	
Plan de Desarrollo Institucional 2011-2016. “Una nueva visión para una nueva época”	
ASEGURAMIENTO DE LA CALIDAD (SISTEMA INTEGRADO DE GESTIÓN)	
Norma Técnica de Calidad en la Gestión Pública. NTCGP 1000:2009	
Proceso de Tecnología de la Información	Proceso de Logística
<ul style="list-style-type: none"> • Caracterización del proceso • PTH02 - Procedimiento para proveer y administrar sistemas de información. V4. • PTH07 - Procedimiento para proveer software específico. V1. • PTP01 – Procedimiento para el análisis, desarrollo y mantenimiento de sistemas de información. V3 	<ul style="list-style-type: none"> • Caracterización del proceso • PLH01 - Procedimiento para la Adquisición de Bienes y Servicios. V14 • PLH21 – Procedimiento para la contratación directa. V4 • ILH01 – Ingreso y actualización de proveedores. V5
GESTIÓN DE PROYECTOS	
<ul style="list-style-type: none"> • Metodología del Marco Lógico. Departamento Nacional de Planeación. 	<ul style="list-style-type: none"> • Guía de los Fundamentos para la Dirección de Proyectos. Guía PMBOK. Cuarta Edición.
ESTÁNDARES DE CALIDAD RELACIONADOS CON TI	
CMMI para Adquisición (CMMI ACQ). V. 1.3	

Tabla 1. Marco de Referencia del Modelo de Gestión para Adquisición de TI, PCJIC.

En ese contexto, el modelo propuesto permitió gestionar un proceso desde la fase de identificación de necesidades tecnológicas hasta la gestión de proveedores y evaluación de propuestas de TI, evidenciando con ello que el modelo se ajusta a los requerimientos del Politécnico.

6. Conclusiones y Trabajos Futuros

Se diseñó y se puso en ejecución un modelo de gestión, que tuviera en cuenta las buenas prácticas en la gestión de proyectos utilizadas por el sector privado, y las condiciones legales, normativas y de calidad del Politécnico Colombiano Jaime Isaza Cadavid.

La experiencia en las etapas iniciales del proceso de implantación del modelo desarrollado permite inferir que la gestión de procesos, esté fuertemente ligada con la cultura de las entidades y con la gestión del cambio, que en cada una de ellas se coordine. Por lo tanto, no depende única ni exclusivamente de la racionalidad del modelo.

El modelo implementado ha permitido incorporar mejoras para optimizar los procesos, pero si las unidades de gestión no son flexibles a los cambios que presenta el entorno, no podrán mejorar continuamente, lo que dificultaría el avance general de la Institución en materia tecnológica.

En cada una de las fases propuestas en el modelo de adquisición de TI, la Institución tuvo en cuenta la importancia de contar con expertos, tanto internos como externos, que sirvieran de apoyo a un proceso efectivo de selección de la tecnología a adquirir.

Referencias

Agudelo M., Elena, Niebles A., Lina María y Gallón L., Luciano. (2005). La Gestión Tecnológica como Herramienta de Planeación Estratégica y Operativa para las Unidades de Información. Revista Interamericana de Bibliotecología, 89 – 114.

Amador, Belkys y Márquez, Alexandra. (2009). Un Modelo Conceptual para Gestionar la Tecnología en la Organización. Revista Espacios, 7 - 21

Benavides Velásquez, Carlos y Quintana García, Cristina. (2007). Un Modelo para la Gestión Estratégica de los Recursos Tecnológicos. Revista Economía Industrial, 195 – 206.

Builes Vélez, Carlos Alberto. (2008). Notas acerca de la Ingeniería de Software. Medellín: Editorial Artes y Letras.

Carroz, Darnila. (2005). Modelo de Gestión Estratégica para el Desarrollo de Capacidades Tecnológicas. Revista Compendium, 5 -19.

Castillo Giraldo, Edwin Marino. (2006). Aseguramiento de la Calidad en el Proceso Delegado de Desarrollo de Software. Tesis de Ingeniería de Sistemas, Universidad EAFIT, Medellín.

COTEC, Fundación Española para la Ciencia y Tecnología. (2010). Análisis de Patrones en el Proceso de Transferencia de Conocimiento y Tecnología. Modelo de Transferencia de Tecnología y Conocimiento. Madrid: Autor.

COTEC, Fundación Española para la Ciencia y Tecnología. (1999). Pautas Metodológicas en Gestión de la Tecnología y de la Innovación para Empresas. Madrid: Autor.

Díaz Spataro, Mariana. (2012). Outsourcing, guía práctica de técnicas y estrategias. Bogotá: Ediciones de la U.

Gay, Aquiles. (1994). La Tecnología, el Ingeniero y la Cultura. Capítulo 1: La Ciencia, la Técnica y la Tecnología. Ediciones Tec. Córdoba, Argentina.

Hernández, René. (2011). Competencias para la Gestión de la innovación en la Formación Gerencial. Revista Formación Gerencial, 217 – 234.

Hidalgo Nuchera, Antonio. (1999). La Gestión de la Tecnología como Factor Estratégico de la Competitividad Industrial. Revista Economía Industrial, 43 – 54.

IDC, International Data Corporation. (2012). Top 10 Predictions. IDC Predictions 2013: Competing on the 3rd Platform. MA: Autor.

Miranda, Juan José. (2010). El Desafío de la Gerencia de Proyectos. Bogotá: MM editores.

Ochoa Ávila, Migdely, Valdés Soa, Mario y Quevedo Aballe, Yovanni. (2007). Innovación, Tecnología y Gestión Tecnológica. Revista Cubana de los Profesionales de la Información y de la Comunicación en Salud, (s/p).

Pantaleo, Guillermo. (2011). Calidad en el Desarrollo de Software. Buenos Aires: Alfaomega.

Pedroza, Álvaro. (2001). Modelo para la Gestión Estratégica de Tecnología. Revista Universidad Eafit, 23 - 37.

Pereña Brand, Jaime. (1996). Dirección y Gestión de Proyectos, (2da.Edición). Madrid: Ediciones Díaz de Santos.

Ramírez González, Leonardo. (2011). Arquitectura Empresarial, mitos y realidades. Encuentro ACIS, XXXI Salón de Informática, Bogotá, 12 – 14 de septiembre, (paper).

Sistema Integrado de Gestión. (2011). CPL07 – Caracterización del Proceso Gestión Logística V.6, Politécnico Colombiano Jaime Isaza Cadavid, Medellín: Intranet Sinapsis.

Sistema Integrado de Gestión. (2011). CPT11 – Caracterización del Proceso Gestión Tecnología de la Información V.6, Politécnico Colombiano Jaime Isaza Cadavid, Medellín: Intranet Sinapsis.

Sistema Integrado de Gestión. (2009). MLH01 – Manual de contratación V.2, Politécnico Colombiano Jaime Isaza Cadavid, Medellín: Intranet Sinapsis.

Sistema Integrado de Gestión. (2011). PLH01 – Procedimiento para la Adquisición de Bienes y/o Servicios – Estudio previo V.14, Politécnico Colombiano Jaime Isaza Cadavid, Medellín: Intranet Sinapsis.

Sistema Integrado de Gestión. (2009). PTH02 – Procedimiento para Proveer y Administrar los Sistemas de Información V.04, Politécnico Colombiano Jaime Isaza Cadavid, Medellín: Intranet Sinapsis.

Sistema Integrado de Gestión. (2011). PTH07 – Procedimiento para Proveer Software Específico V.01, Politécnico Colombiano Jaime Isaza Cadavid, Medellín: Intranet Sinapsis.

Software Engineering Institute. (2011). CMMI for Acquisition (CMMI-ACQ) V. 1.3, Carnegie Mellon University, Pittsburgh.

Vega González, Luis Roberto. (2006). Modelo del Ciclo de Vida de un Proyecto de Gestión Tecnológica y Vinculación en un Centro de I+D Universitario. I Congreso Iberoamericano de Ciencia, Tecnología, Sociedad e Innovación, Universidad Nacional Autónoma de México, Ciudad de México, 19 - 23 de julio, (paper).

Notas biográficas



Rafael David Rincón Bermúdez obtuvo el título de Maestría en Sistemas de Calidad del ITESM, Campus Monterrey de México, y se desempeña como profesor Titular del Departamento de Sistemas de la Universidad Eafit de Medellín, Colombia. Consultor en Mejora de Procesos de Negocio, Implementación de Sistemas, Modelos y Estándares de Calidad de Software, como CMMI.



Guadalupe Peláez Ramírez es Tecnóloga en Sistematización de Datos y actualmente, estudiante de pregrado de Ingeniería Informática del Politécnico Colombiano Jaime Isaza Cadavid; igualmente se ha desempeñado como auxiliar de docencia en el área de Ingeniería de Software y docente de cátedra del proyecto de los Consultorios Tecnológicos en el Politécnico. En su trabajo como analista de datos en la Oficina Asesora de Planeación, su labor se fundamenta en la automatización de gestión de procesos y monitoreo de indicadores del Sistema de Calidad.



Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Herramienta para la Automatización de la Revisión Sistemática

Edgar Felipe Uribe Rayas

Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
edgar.uribe@cimat.mx

Jorge Alexis Márquez del Real

Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
jorge.marquez@cimat.mx

J. Guadalupe Uribe Dévora

Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
guadalupe.uribe@cimat.mx

Claudia Valtierra Alvarado

**Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
claudia.valtierra@cimat.mx**

Jezreel Mejia Miranda

**Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
jmejia@cimat.mx**

Resumen: La realización de una revisión sistemática provee a los investigadores las bases para obtener conocimiento científico mediante la identificación de artículos científicos relevantes, los cuales contengan las tendencias en investigación y experimentación acerca de un tema en específico. Existen varios estudios acerca de cómo implementar una revisión sistemática, sin embargo, estas investigaciones solo indican los pasos que conforma el protocolo de la revisión sistemática, sin proporcionar alguna herramienta o aplicación de software que soporte la automatización de este protocolo, dando como resultado que los investigadores dediquen una buena cantidad de tiempo en la detección y búsqueda de artículos científicos relevantes de manera manual. Por lo tanto, en este artículo se presenta la propuesta de una herramienta para la automatización del protocolo de la Revisión sistemática, con la finalidad de reducir el tiempo en el desarrollo del protocolo de la Revisión Sistemática.

Palabras clave: Revisión Sistemática, Protocolo, Herramientas.

A Tool to Automate the Systematic Review

Abstract: Systematic Reviews provide to the researchers the base for obtaining scientific knowledge through of the identification of relevant scientific articles, which contain research tendencies and experimentation about a specific topic. There are several studies about how to implement a systematic review, however these studies only indicate the systematic review protocol and do not provide some software tool or application to support the automation of this protocol, giving as a result that the researchers expend large amounts of time to search and detect relevant scientific articles manually. Therefore, in this paper is proposed a tool to automatize the systematic reviews protocol, in order to decrease the time to perform the systematic review protocol.

Keywords: Systematic Review, Protocol, Tools.

1. Introducción

La inclusión de revisiones sistemáticas en la ingeniería de software provee un medio efectivo para la identificación y evaluación de la literatura e investigaciones existente en un dominio específico. Esta actividad es considerada como el primer paso a llevar a cabo en la obtención de conocimiento, entendimiento y delimitación de un tema de investigación científico (Mian, Conte, Natali, Biolchini, & Travassos, n.d.). Varios análisis sobre revisiones sistemáticas y su aplicación a la ingeniería del software, han sido realizados (Group, 2007; Jørgensen & Shepperd, 2007; Kitchenham & Dybå, 2004; Mian et al., n.d.). Estos trabajos analizan las actividades que se realizan en una revisión sistemática y proponen nuevos pasos o actividades para llevar a cabo este protocolo, sin embargo, y a pesar de sentar las bases para el establecimiento del protocolo de la revisión sistemática, hasta el momento se han detectado pocas herramientas existentes, que de manera eficiente soporten la automatización de las actividades de una revisión sistemática de manera completa. La mayoría de herramientas como Mendeley (Ltd, 2013), RefWorks (RefWorks, 2013), JabRef (Team, 2013), EndNote (Thomson, 2013), que sirven de apoyo al investigador se enfocan

principalmente en la elaboración de citas bibliográficas, administración de estudios encontrados y en la interacción con algunas bibliotecas digitales.

Por lo tanto, el objetivo principal de esta investigación es presentar el desarrollo de una herramienta o aplicación de software que permite la automatización del protocolo que sigue una revisión sistemática. El artículo está estructurado de la siguiente manera: en la sección 2, se presenta las fases y las actividades del protocolo de la revisión sistemática; en la sección 3 se analizan herramientas existentes para establecer una comparativa y detectar mejoras para ser incluidas en la herramienta propuesta; en la sección 4, se presenta la propuesta de la herramienta para la revisión sistemática; en la sección 5, se presenta el caso de estudio en los que se aplica la herramienta y finalmente, la sección 6 se presentan las Conclusiones y trabajos futuro posibles.

2. Protocolo de la Revisión Sistemática

De acuerdo a la definición de Kitchenham (Group, 2007), la revisión sistemática es un método que permite a los especialistas obtener resultados relevantes y cuantificables, esto lleva a la identificación, selección y producción de evidencias basadas en la investigación de un tópico en particular (Mian et al., n.d.). Para establecer las fases y pasos que debería contener la herramienta que se propone, fue necesario realizar un análisis de diferentes autores acerca de cómo implementar una revisión sistemática. Pino(Espa, 2006) señala las actividades que se deben realizar para llevar a cabo una revisión sistemática. Kitchenham, Dyba y Jorgensen (Kitchenham & Dybå, 2004) señalan la importancia que adquiere la práctica de revisiones sistemática en la ingeniería del software y explican de manera general como se implementa una revisión, de donde se puede extraer el método a implementar en la herramienta. Un ejemplo de selección de estudios primarios, y la importancia de la estadística se muestra también en el trabajo realizado por Jorgensen y Shepperd (Jørgensen & Shepperd, 2007). Un enfoque de

aplicación de las revisiones sistemáticas y los pasos a seguir en la Ingeniería del Software es presentado por Mian, Conte, Natali, Biolchini y Travassos (Mian et al., n.d.), establece el proceso de las revisiones sistemáticas adecuado a la ingeniería del software.

De la investigación realizada se obtuvieron las fases principales y los pasos que estas deben integrar para una revisión sistemática, las cuales son las siguientes:

- Planificación de la revisión: la primer fase, en esta etapa se identifica la investigación, se establecen los objetivos y se realizan las siguientes actividades:
 - Elección de nombre de la investigación.
 - Formulación de las cadenas de búsqueda.
 - Selección de fuentes de búsqueda de estudios y establecer los criterios de selección de estudios primarios y secundarios.
- Realizar la revisión sistemática: en esta segunda fase se ejecutan las siguientes actividades:
 - Ejecución de la búsqueda.
 - Evaluación de la calidad de los estudios.
 - Revisión de los estudios seleccionados.
 - Extracción de la información.
 - Documentos de extracción de datos.
 - Ejecución de la extracción.
- Reporte de la revisión: tercera y última fase en la que se resume y analizan los resultados utilizando métodos estadísticos, las actividades que se contemplan son:
 - Cálculo estadístico.
 - Presentación de resultados.

3. Herramientas existentes

Una vez establecidas las fases del protocolo y las actividades contenidas en cada una de ellas, se realizó un estudio sobre las herramientas existentes que favorecen la ejecución de una revisión sistemática. Se encontró que si bien existen aplicaciones que brindan soporte en algunos aspectos, éstas no cubren completamente el proceso que conlleva la realización de una revisión sistemática, entre las herramientas que se encontraron se citan las siguientes:

- Mendeley (Ltd, 2013), es una herramienta para la elaboración de citas bibliográficas, control de estudios agregados de manera manual y cuenta con un plugin llamado webimporter, el cual se utiliza para agregar la funcionalidad de descargas de la aplicación a los distintos navegadores web, sin embargo, las búsquedas se siguen realizando de manera manual, lo que conlleva a que los investigadores sigan dedicando un considerable lapso del tiempo de la revisión sistemática a la búsqueda de artículos.
- RefWorks (RefWorks, 2013), es una aplicación online de paga, permite un periodo de prueba de 30 días gratuitos. La herramienta funciona como un gestor en línea de artículos y referencias, artículos pero de forma manual, es decir el investigador debe tener los artículos o estudios digitales en su computadora e importarlos a la aplicación que se mueve sobre la plataforma web, en palabras sencillas RefWorks sirve para creación de citas bibliográficas y administración de los artículos.
- JabRef (Team, 2013) es una aplicación open source que permite entre otras características, al igual que las dos anteriores herramientas, la administración de artículos y referencias, así como la inclusión de éstas en investigaciones.
- EndNote (Thomson, 2013) es una herramienta para administración y publicación de bibliografías, lo interesante de la herramienta es que permite aplicar hasta 5000 estilos de citas bibliográficas.

Cabe decir que las 4 aplicaciones permiten la creación de una base de datos de artículos alojados en la computadora de los investigadores, además,

permiten la búsqueda de datos de un artículo en internet, pero con el fin de crear citas bibliográficas, y aunque son de una gran utilidad no brindan el soporte necesario para la realización de una revisión sistemática.

Como resultado del análisis, la Tabla 1 muestra que los pasos identificados del protocolo de la revisión sistemática no son incluidos en las herramientas analizadas.

Herramienta Elementos	Mendeley	RefWorks	JabRef	EndNote
Citas bibliográficas	Si	Si	Si	Si
Búsquedas en internet	Si	Si	Si	Si
Gestión de estudios	Si	Si	Si	Si
Asignación de Prioridad	No	No	No	No
Búsquedas en bibliotecas digitales	Si	Si	No	Si
Extracción automática de información	Si	Si	Si	Si
Elaboración de Plantillas	No	No	No	No
Elaboración de gráficos para análisis	No	No	No	No
Selección de estudios primarios.	No	No	No	No

Tabla 1. Comparación de funciones de herramientas similares.

Como puede observarse, no existe alguna herramienta que cubra y automatice las actividades identificadas para el protocolo de la revisión sistemática, y que en el desarrollo de la herramienta deben ser establecidos como funcionalidad principal.

4. Propuesta de Herramienta para la Revisión Sistemática

En esta sección se describen los módulos que integran la herramienta propuesta. Para facilitar el uso de la herramienta, ésta se divide en módulos de fácil acceso y comprensión, para proporcionar a los investigadores un ambiente amigable y práctico. A continuación se describen los módulos de la aplicación, los principales son mostrados en la Figura 1, y en la Figura 2, y están basados en la identificación de las fases y actividades obtenidas en la sección 2.

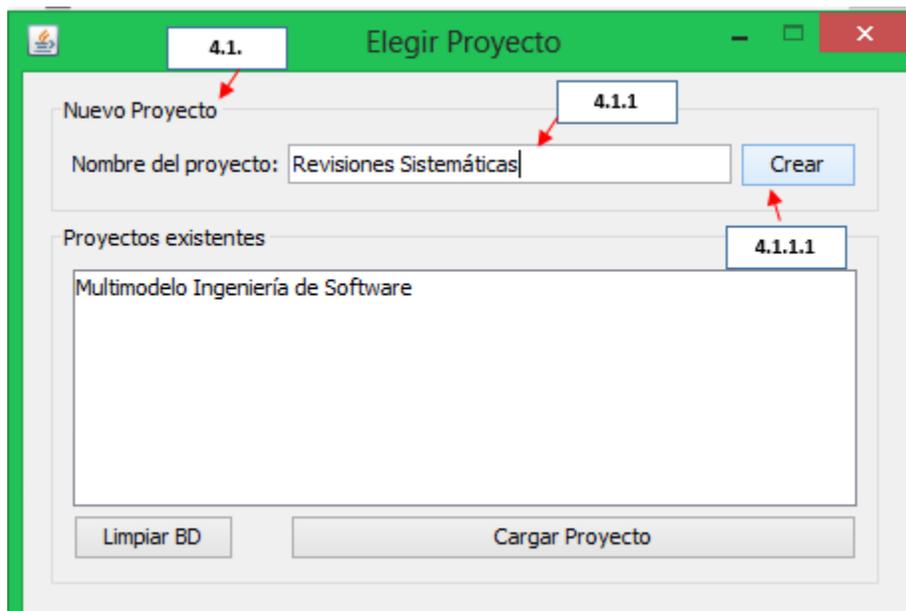


Figura 1. Módulo de Soporte a la Planificación de la Revisión.

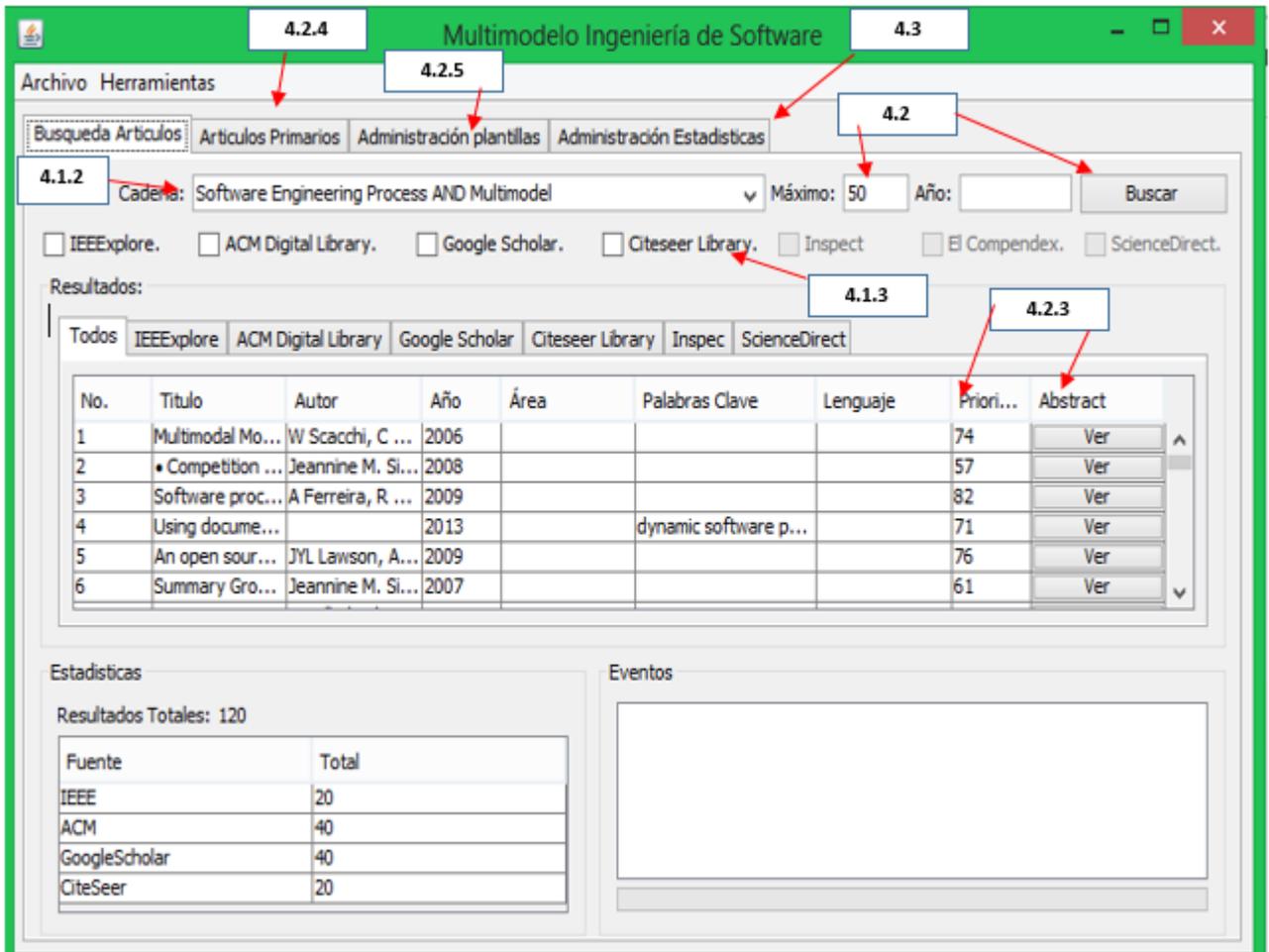


Figura 2. Principales Módulos de la Herramienta para la Automatización del Protocolo de la Revisión Sistemática.

4.1 Planificación de la revisión.

La herramienta cubre esta fase del protocolo mediante los siguientes módulos:

4.1.1 Elección de nombre de la investigación:

se establece el entorno para realizar esta función mediante la ventana de nuevo proyecto que permite la creación de un proyecto de investigación o revisión sistemática, a continuación se agrega una breve descripción de este módulo.

4.1.1.1 Creación de Un proyecto

Al crear un nuevo proyecto se le asigna un nombre de acuerdo al tópico de la investigación que se realiza, este proyecto de manera automática, se agrega a la base de datos de la aplicación, todas las búsquedas, estudios, archivos, plantillas, estadísticas y descargas se almacenen para su posterior uso en la base de datos. La principal función de este módulo es proveer orden en la administración de estudios y revisiones, al tener ordenados en la base de datos los proyectos y sus elementos, simultáneamente se crea una estructura de carpetas que permite tener ordenados artículos, plantillas y estadísticas de cada estudio permitiendo un perfecto control sobre la ubicación de cada elemento de la investigación.

4.1.2 Formulación de las cadenas de búsqueda:

El motor de búsqueda que se implementa en la herramienta está basado en los criterios utilizados por las principales bibliotecas digitales, esta actividad se realiza en base al módulo de la herramienta descrito enseguida.

4.1.2.1 Módulo de búsqueda de artículos.

Permite la entrada de cadenas de búsqueda formadas por los investigadores, las cadenas pueden estar formadas por los principales tópicos de una investigación, luego el sistema las convierte a expresiones regulares para realizar la extracción de estudios en las fuentes, los datos por los que se puede buscar artículos o formar cadenas de búsqueda son: a) Título, b) Palabras principales del tópico de la investigación, c) Tópicos enlazados con conectores lógicos booleanos AND, OR y NOT, d) Autor, e) Sinónimos y abreviaturas.

4.1.3 Selección de fuentes de búsqueda de estudios y establecer los criterios de selección de estudios primarios y secundarios:

La herramienta cuenta con un módulo de búsqueda de estudios el cual permite la elección de bibliotecas digitales, el módulo que realiza esta función es el siguiente:

4.1.3.1 Bibliotecas Digitales:

Este módulo incrustado en la pantalla de búsqueda permite selección y la extracción de estudios de las siguientes fuentes: 1) IEEExplore, 2) ACM Digital Library, 3) Citeseer Library, 4) Inspect, 5) EICompindex, 6) ScienceDirect, 7) Scholar Google.

4.1.3.2 El criterio de selección de estudios y proceso de selección de estudios.

El sistema favorece la selección de estudios, y considera los siguientes factores: a) Año de publicación, b) Incidencia de autores en el tema de investigación, c) Incidencia de palabras clave dentro del abstract de cada documento.

4.2 Realizar la revisión sistemática

La herramienta permite la automatización de esta segunda fase mediante un conjunto de módulos que realizan las funciones requeridas y que se describen a continuación:

4.2.1 Número máximo de estudios a buscar por biblioteca.

Esta opción permite establecer un máximo de estudios ordenados por prioridad que se busquen y mostraran en cada librería.

4.2.2 Ejecución de la búsqueda.

Cuando se han introducido la cadena de búsqueda, se eligió un máximo de estudios y se han elegido las fuentes para extracción de los estudios, la herramienta cuenta con un botón que inicia la búsqueda en las fuentes seleccionadas este botón también permite detener una búsqueda e iniciar una nueva con distintos parámetros.

4.2.3 Calidad y revisión de los estudios seleccionados.

La evaluación de los estudios se lleva a cabo en base a la priorización de estudios primarios, los módulos que realizan la actividad relacionada a la calidad de los estudios son:

4.2.3.1 Resultados:

En esta pantalla la herramienta permite observar los resultados emanados de una búsqueda, los resultados se muestran:

- Todos: muestra todos los resultados encontrados.
- Por biblioteca Digital: mediante pestañas que muestran los resultados encontrados en una biblioteca en específico.

Los campos que se muestran de cada estudio cuando las bibliotecas y los documentos permiten extraerlos son: a) Título, b) Autor, c) Año, d) Área, e) Palabras clave, f) Lenguaje, g) Prioridad, h) Abstract.

4.2.3.2 Evaluación de la calidad de los estudios:

El sistema permite mostrar el resumen (Abstract) de cada documento mediante la opción ver para constatar de manera visual, si el estudio corresponde a la prioridad asignada. La selección de estudios primarios basada en la prioridad del documento encontrado y la evaluación mediante la lectura y la visualización del abstract, si estos dos criterios coinciden a la vista del investigador el estudio

se puede considerar primario o secundario en caso de no cumplir con las expectativas del proyecto de la revisión.

4.2.4 Extracción de la información.

El módulo base que realiza las funciones de extracción y síntesis de la información de los estudios de esta actividad es el de administración de artículos primarios que se describe enseguida.

4.2.4.1 Módulo de administración de artículos primarios:

La herramienta permite ordenar y mantener un registro de los artículos primarios del proyecto, el ordenamiento se puede mostrar por los siguientes criterios:

- Por archivo asignado.
- Sin archivo asignado.
- Por plantilla.

Identificado un estudio como primario, esta funcionalidad permite dos opciones:

- Descargar el artículo de forma automática.
- Agregar artículo de forma manual.

4.2.5 Documentos de extracción de datos y ejecución de la extracción.

Herramienta ofrece un sistema de creación y administración de plantillas estandarizadas para reportar la información de estudios, se toman en cuenta para la conformación de una plantilla los siguientes puntos:

- Una plantilla puede contener la información de un documento o de varios documentos.
- Los campos de la plantilla son personalizados, es decir el investigador los puede definir.
- El sistema de Administración de plantillas permite la creación, modificación y eliminación de plantillas.
- El sistema de Administración de plantillas permite la visualización, modificación, eliminación y exportación de los artículos que conforman una plantilla.

4.3 Reporte de la revisión

Para esta tercera fase, la herramienta permite la elaboración de reportes en base a las plantillas en conjunto con el módulo de análisis que permite realizar gráficas estadísticas. El objetivo es proveer a los investigadores un sistema que les permita el análisis de los estudios que han seleccionado y obtener resultados en base a las plantillas de reporte que se han creado. Para agregar una gráfica primero se selecciona la plantilla de la cual se va a graficar el criterio u criterios que se van a evaluar, por defecto se selecciona la opción general y en este caso se grafican los elementos de todas las plantillas existentes, también es posible seleccionar solo una plantilla, automáticamente se muestran los campos generales de las plantillas, aunque también se puede seleccionar de manera manual los criterios personalizados en cada plantilla. La visualización de las gráficas se realiza en el mismo módulo mediante la opción ver y para el control de documentación de la investigación también se pueden exportar a un documento .docx, al igual que los reportes de las plantillas.

4.3.1 Sistema de referencias

El sistema cuenta con un módulo de adición de citas bibliográficas, las cuales se integran a las plantillas de reporte o si se desea se pueden importar desde el menú herramientas lo que permite exportar las citas bibliográficas en varios formatos, entre los cuales se encuentran entre otros:

- IEEE
- AMERICAN MEDICAL ASOCIATION (AMA).
- Harvard Reference format 1 (autor-fecha).
- Nature.
- American Sociological Association.

5. Caso de estudio

Actualmente la herramienta se aplica para realizar la revisión sistemática en la investigación de 4 tesis de estudiantes en el Centro de Investigación en Matemáticas CIMAT Unidad Zacatecas, con lo cual se espera tener una visión del comportamiento de la herramienta. Hasta el momento se han introducido cadenas de búsqueda en la herramienta para obtener archivos de las diversas fuentes. Con ésta se obtiene un número de estudios, así como, sus datos y archivos y una prioridad asignada según el criterio de la herramienta, en contraste para comprobar su efectividad se introduce la misma cadena en cada una de las bibliotecas digitales en donde se encontraron los estudios en las diferentes bases de datos que maneja la herramienta y se ha constatado la efectividad pues se logran detectar los mismos estudios y con similar prioridad lo que indica que los criterios aplicados son correctos en un alto porcentaje. Cabe resaltar, que las búsquedas manuales que se han realizado en forma manual se han reducido hasta en un 70% de cuando se hacía manualmente, debido a la automatización de las búsquedas, descargas y sistema de monitoreo que permite la herramienta.

6. Conclusiones y Trabajos Futuros

El uso de la herramienta agiliza el proceso de realización de una revisión sistemática, hasta este momento con las pruebas que se han realizado, las

búsquedas con la herramienta proveen resultados factibles y acordes con las búsquedas manuales. La herramienta brinda un soporte idóneo para la elaboración de reportes, se han buscado herramientas similares pero ninguna realiza las funciones que la aplicación objeto de esta investigación ejecuta. En conclusión el equipo de trabajo de la investigación de desarrollo de la herramienta espera colaborar con este elemento en la investigación para facilitar el trabajo de los investigadores y mejorar el uso de fuentes literarias, aunque al final el criterio del investigador es esencial. Como trabajo futuros implica probar la aplicación en más casos de estudio para constatar su efectividad y tener un rango más amplio de resultados del uso de ésta, se planea de acuerdo a los resultados obtenidos una nueva versión que pueda incluir mejoras y correcciones que se puedan detectar, ya está en proceso del desarrollo de un plugin que permita integrar la herramienta con los distintos procesadores de texto para ampliar la funcionalidad y uso de la herramienta, aunque en un principio sea sólo en la inserción de citas bibliográficas, al final se buscará una mayor interacción con herramientas para brindar al investigador un marco de trabajo ideal.

Referencias

Espa, R. (2006). Española de Calidad e Ingeniería del Software.

Group, S. E. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering.

Jørgensen, M., & Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies, 33(1), 33–53.

Kitchenham, B. A., & Dybå, T. (2004). Evidence-based Software Engineering.

Ltd, M. (2013). Mendeley Ltd. Retrieved November 01, 2013, from <http://www.mendeley.com/>

Mian, P., Conte, T., Natali, A., Biolchini, J., & Travassos, G. (n.d.). A Systematic Review Process for Software Engineering.

RefWorks. (2013). Refworks. Retrieved October 30, 2013, from <http://www.refworks.com/>

Team, J. D. (2013). JabRef[desktop]. Retrieved November 02, 2013, from <http://jabref.sf.net>.

Thomson, R. (2013). EndNote[desktop]. Reuters Thomson. Retrieved from <http://endnote.com>.

Notas biográficas:



Edgar Felipe Uribe Rayas es Ingeniero en Sistemas Computacionales, egresado de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés es el desarrollo de software basado en procesos, modelos de calidad, la seguridad informática y la arquitectura de software, ha desarrollado varios sistemas de software y sitios web mediante el uso de varios lenguajes de programación.



Jorge Alexis Márquez del Real es Ingeniero en Sistemas Computacionales, egresado de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés es la y arquitectura de software, la aplicación de modelos matemáticos a los procesos de software, ha desarrollado varios sistemas de software y sitios web mediante el dominio de varios lenguajes de programación.



J. Guadalupe Uribe Dévora es Ingeniero en Sistemas Computacionales, egresado de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés es el desarrollo de software basado en procesos, la arquitectura de software y los modelos de calidad, ha desarrollado varios sistemas de software y sitios web mediante el dominio de varios lenguajes de programación.



Claudia Valtierra Alvarado es una Ingeniero en Sistemas Computacionales, egresada de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés son los procesos de software, el estudio de la caracterización de organizaciones de desarrollo y el diseño web, ha desarrollado varios sistemas de software y sitios web mediante el dominio de varios lenguajes de programación.



Jezreel Mejia Miranda es doctor en Informática por la Universidad Politécnica de Madrid (UPM), España, donde se le concedió la nota máxima, Cum Laude, y mención de "Doctorado Europeo". Realizó una estancia de investigación para obtener el doctorado europeo en la Universidad Fernando Pessoa en Oporto, Portugal. Previamente, en el Instituto Tecnológico de Orizaba, Veracruz, cursó la maestría en Ciencias de la Computación y la licenciatura en Informática. Es miembro del grupo de investigación Cátedra de Mejora de Procesos Software en el Espacio Iberoamericano (MPSEI), donde participa en proyectos internacionales de investigación con entidades educativas y de gobierno (Instituto Tecnológico de Orizaba; Instituto Regional de Zacatecas; Facultad de Informática de la UPM) y de vinculación con la industria (clúster de empresas de desarrollo de software en Zacatecas). Asimismo, es miembro del comité científico de diversos congresos internacionales como: CISTI (2009-2013),

CERMA (2009-2013), del Coloquio de Investigación Multidisciplinaria del Instituto Tecnológico de Orizaba (2011) y del Infonor Chile 2012 y de la revista internacional RISTI (2010-2013). Ha publicado diversos artículos técnicos en temas relacionados con la gestión de proyectos, entornos multi-modelo, modelos y estándares de calidad y temas relacionados en entornos outsourcing. También ha participado en proyectos de la empresa multinacional everis consulting. Actualmente, el Dr. Jezreel Mejía Miranda es investigador del Centro de Investigación en Matemáticas, A.C. (Cimat), Unidad Zacatecas, en el área de Ingeniería de Software. También forma parte del equipo oficial de traducción al español del libro CMMI-DEV v1.2 y 1.3, versiones reconocidas por el prestigioso Software Engineering Institute (SEI) de la Carnegie Mellon University. Como investigador, sus áreas de interés son: entornos multi-modelo, gestión de proyectos software, modelos y estándares de calidad (CMMI, ISO, TSP, PSP, etc.), metodologías ágiles, métricas, mejora de procesos en entornos outsourcing y entornos de desarrollo tradicional. Cuenta con certificación en CMMI e ISO 20000.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Identificando Hallazgos de Mejora en Pymes de TI Utilizando un Modelo Ontológico para CMMI-DEV v1.3

J. Guadalupe Uribe Dévora

Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
guadalupe.uribe@cimat.mx

Edgar Felipe Uribe Rayas

Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
edgar.uribe@cimat.mx

Jorge Alexis Márquez del Real

Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068

Zacatecas, Zacatecas, México
jorge.marquez@cimat.mx

Claudia Valtierra Alvarado

Centro de Investigación en Matemáticas, CIMAT A.C. Unidad
Zacatecas
Av. Universidad No. 222
Fracc. La Loma
C.P. 98068
Zacatecas, Zacatecas, México
claudia.valtierra@cimat.mx

Resumen: Uno de los principales requisitos que deben cumplir las pymes de desarrollo de software es adoptar un estándar o modelo para garantizar la calidad de sus productos. Sin embargo, este tipo de organización, carece o no tienen procesos bien definidos. En este contexto implementar un modelo o estándar de calidad para estas organizaciones enfrenta algunos obstáculos como: gran desconocimiento sobre éstos, su adopción representa una curva de aprendizaje prolongado, las certificaciones de estos modelos o estándares para las pymes y su personal representan costos muy elevados. Por lo tanto, en este artículo se presenta una herramienta ontológica basada en el Modelo CMMI-DEV v1.3. El objetivo principal es proveer a las pymes de una herramienta que contraste sus procesos actuales obtenidos a través de un método para la extracción del conocimiento tácito versus los procesos de CMMI-DEV v1.3 para detectar posibles hallazgos de mejora e implementar mejoras en sus procesos organizacionales.

Palabras clave: CMMI-DEV v1.3, Modelo Ontológico, Herramienta Ontológica, Extracción del Conocimiento.

Identifying Improvement Findings in IT SMEs Using an Ontological Model for CMMI-DEV v1.3

Abstract: One of the main requirements that software development SMEs must fulfill is to adopt a standard or model that guarantees the quality in their software products or services. However, this kind of organization lacks of well-defined processes. In this context, implementing a quality model or standard for them presents some difficulties: lack of knowledge about these, its adoptions takes a prolonged learning curve, the certification of these for SMEs

and their staff have high costs. Therefore, this paper presents a tool based ontological Model CMMI-DEV v1.3. The main objective is to provide SMEs a tool that contrast their current processes obtained through a method to extract of knowledge tacit into CMMI-DEV v1.3 processes to identify improvement findings for implementing them in their organizational processes.

Keywords: CMMI-DEV. v1.3, Ontological Model, Ontological Tool, Knowledge Extraction.

1. Introducción y Motivación

Actualmente la calidad en productos y servicios de software es un factor crucial para el éxito de cualquier organización. Un alto porcentaje de empresas que generan fuentes de ingresos en la industria del desarrollo del software según la investigación de Pino, Garcá y Piattini (Espa, 2006), son las llamadas pymes. Estas organizaciones se han dado cuenta de la necesidad de implementar procesos que permitan incrementar la calidad de los productos y/o servicios de software que generan (Muñoz, Mejía, 2013). Sin embargo, un aspecto en común que comparten este tipo de organizaciones son la falta de procesos bien definidos. Lo que conlleva muchas veces que su entorno de trabajo sea caótico y muy cambiante (Muñoz, Mejía, 2013).

Una solución en este tipo de organizaciones es la adopción de un modelo o estándar de calidad, sin embargo, esto implica algunos obstáculos para estas organizaciones como: la adopción de un modelo o estándar representa una curva de aprendizaje prolongado, las certificaciones de éstos en las pymes y su personal tienen costos elevados, lo que significa una barrera difícil de cubrir, esto aunado a la resistencia al cambio, tal y como lo mencionan los estudios realizados en México, por Muñoz y Mejía (Muñoz, Mejía,2013), y en Brasil, por Santos y Montoni (Santos et al., 2007).

Por lo tanto, el objetivo principal es proveer a las organizaciones de una herramienta que de manera eficaz les permita contrastar sus procesos y actividades de desarrollo, hacia el modelo de Integración de Capacidad y Madurez para Desarrollo (CMMI-DEV v1.3) e identificar los hallazgos de mejora de los procesos detectados en la organización para implementar mejoras con respecto al modelo CMMI-DEV v1.3.

La estructura del artículo está dividida de la siguiente manera: en la sección 2 se describe el desarrollo de la ontología de CMMI-DEV v1.3; la sección 3 se describe la herramienta basada en el modelo ontológico de CMMI-DEV v1.3; la sección 4 aborda la extracción del conocimiento tácito de la organización, la sección 5 está dedicada a la aplicación de la herramienta ontológica, en la sección 6 se presenta el caso de estudio, la sección 7 define las conclusiones y el trabajo futuro a realizar.

2. Ontología de CMMI-DEV v1.3

En esta sección antes de presentar la herramienta ontológica, se muestra la construcción de la ontología para el modelo CMMI-DEV v1.3. Este modelo permite evaluar la madurez de los procesos de software dentro de las organizaciones (Report, 2003) e integra una estructura compleja que requiere de un estudio minucioso (Halit Soydan, 2012). Para comprender el desarrollo del modelo ontológico, esta sección se divide en tres fases principales: el análisis de literatura y estudios previos, la propuesta de la herramienta, y el resultado obtenido del modelo ontológico.

2.1 Estudios previos

La primera fase del desarrollo del modelo ontológico, es analizar varias investigaciones acerca del desarrollo de una ontología que sobre el modelo han sido realizadas (Halit Soydan, 2012; Noy & Mcguinness, 2005; Soydan & Kokar, n.d.). La principal diferencia que se establece en esta investigación es que los estudios previos detectados para la construcción de la ontología para el

modelo CMMI-DEV v1.3 están enfocados en la industria y organizaciones de desarrollo en general, mientras que este estudio se enfoca de manera especial en las empresas catalogadas como pymes. Otra diferencia importante, es que la aplicación de esta ontología utiliza los procesos establecidos con el método de extracción de conocimiento tácito de la organización (Muñoz, Mejía, 2013).

2.2 Propuesta de herramienta basada en el modelo ontológico de CMMI-DEV V1.3

Como lo menciona Soydan (Halit Soydan, 2012) una de los principales problemas al desarrollar un modelo ontológico de CMMI-DEV v1.3 es que se debe tomar en cuenta que existen varias maneras para que una empresa alcance un nivel de calidad, por lo que no se puede manejar una inferencia de nivel fija, en cambio deben de tomarse en cuenta las posibles variantes que pueden ocurrir al introducir la información. Una vez establecido y analizado la estructura, y componentes desde la bibliografía oficial de CMMI-DEV V1.3, se estableció el modelo ontológico viable y los principales elementos y sus relaciones mediante una estructura taxonómica que se muestra en la Figura1.

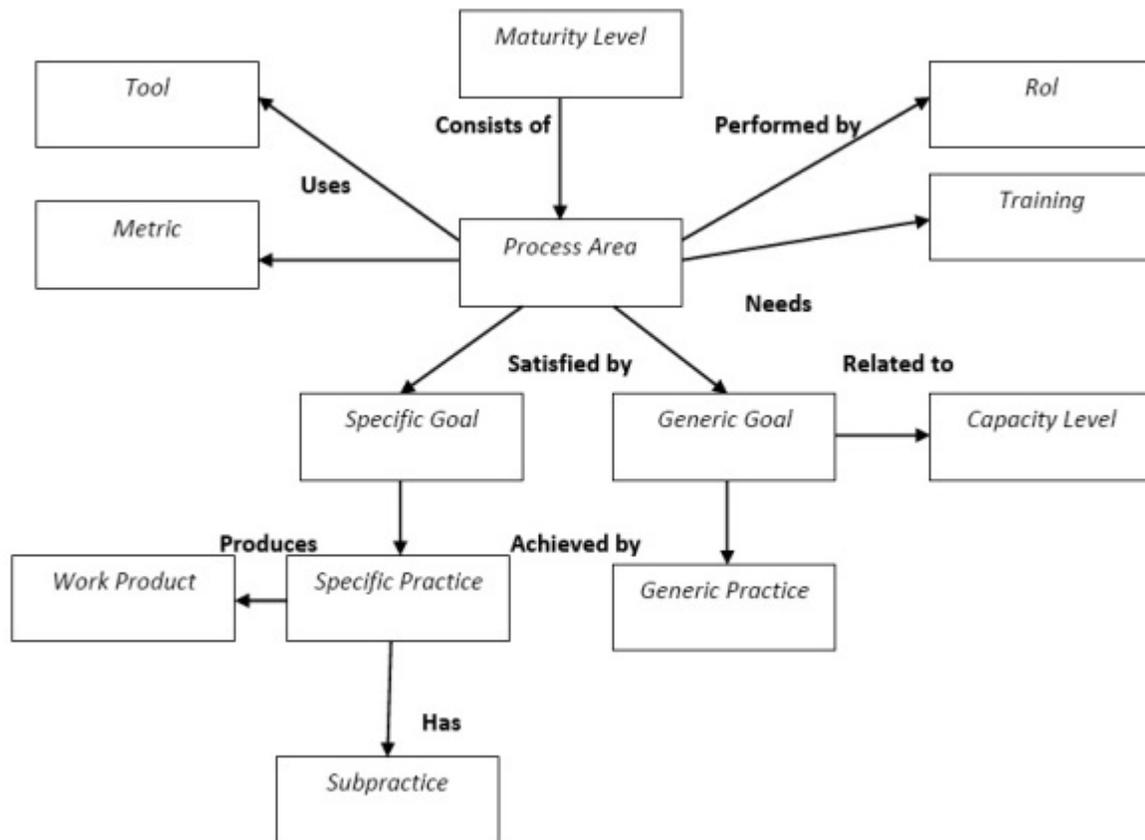


Figura 1. Modelo taxonómico de la ontología de CMMI-DEV v1.3

2.3 Modelo ontológico de CMMI-DEV V1.3

Establecido el modelo taxonómico, en la tercera fase se realizó un modelo ontológico utilizando la herramienta Protégé (Halit Soydan, 2012) y en base al API del lenguaje ontológico OWL (Halit Soydan, 2012) con este propósito en base al conocimiento del análisis de investigaciones previas en la materia acerca de este tópico (Halit Soydan, 2012) se desarrolló una ontología base que incluye las clases, relaciones, restricciones que conforman el modelo CMMI-DEV v.1.3. Las clases del modelo ontológico resultado, se muestra a continuación en la Figura 2:

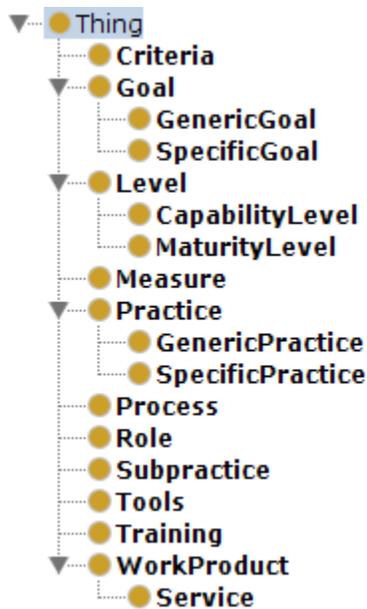


Figura 2. Clases de la ontología de CMMI-Dev V1.3 generadas en Protégé.

3. Construcción de la aplicación ontológica

Al definirse la taxonomía, relaciones y principales elementos de acuerdo a la estructura del modelo ontológico para CMMI-DEV v1.3, la siguiente actividad fue desarrollar una herramienta basada en la ontología. Para el desarrollo de la ontología se utilizó el lenguaje OWL(Onthological Web Language)(Noy & Mcguinness, 2005), para la interfaz se utilizó el lenguaje java. La herramienta permite introducir la información detectada después de la aplicación del método de extracción del conocimiento tácito (Muñoz, Mejía, 2013). Como resultado la herramienta permite observar con que elementos del modelo se cumple, cuales son los que faltan en el proceso según el modelo de calidad de CMMI-DEV v1.3, habilitando al usuario observar posibles mejoras en los procesos de las organizaciones. La Figura 3 muestra la ventana principal de la herramienta ontológica.

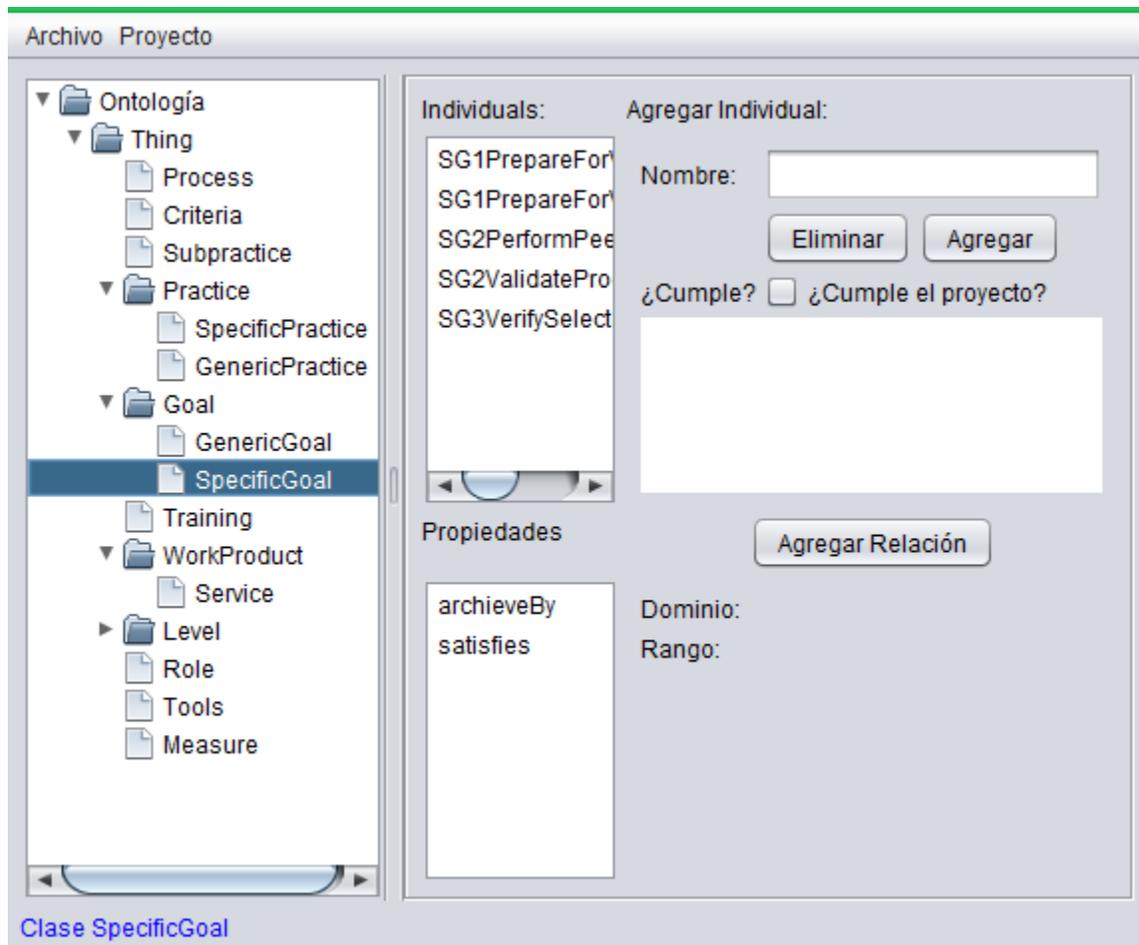


Figura 3. Herramienta de software creada en base a la ontología de CMMI-DEV v1.3

Entre otras características, la herramienta permite cargar ontologías ya realizadas y permite el análisis de las relaciones entre los distintos elementos.

4. Método para la Extracción del Conocimiento Tácito

Antes de utilizar la herramienta en base al modelo ontológico de CMMI-DEV v1.3 descrita en la sección anterior, se requiere tener la información de los procesos que se realizan en una organización.

En esta sección, se presenta el método utilizado para la extracción del conocimiento tácito (Muñoz, Mejía,2013). El método permite obtener el conocimiento acerca del entorno de la organización y los procesos que se realizan aún y cuando no exista un registro formal de las actividades que se llevan a cabo para el desarrollo de software. Dentro de las principales actividades que marca el método utilizado son: Identificación de prácticas internas, Análisis de la información, Diccionario de términos y, Trazabilidad de actividades y procesos. A continuación se describe cada una de estas actividades.

4.1 Identificación de prácticas internas:

Para desarrollar esta actividad se aplica un cuestionario basado en el modelo CMMI-DEV v1.3, respecto al cual se contrasta la información obtenida. El cuestionario es aplicado a los 3 mandos implicados en la estructura organizacional en una pyme, estos son: el mando gerencial, de dirección de proyectos de desarrollo y el operacional que es el equipo de trabajo. Para la aplicación del cuestionario se concretan entrevistas con los involucrados de los diferentes niveles de la organización y se graba el audio que se obtiene de éstas para su posterior análisis. El cuestionario se divide en 9 bloques que contienen una serie de preguntas, cada bloque abarca varios de los procesos de CMMI-Dev v1.3, los bloques que conforman la entrevista son: Planificación de Proyecto, Seguimiento de Proyecto, Identificación de Riesgos, Requisitos de Proyecto, Satisfacción de los clientes, Gestión de Proveedores, Formación del Equipo de Desarrollo, Control de Calidad y Gestión de la Configuración.

4.2 Análisis de la información:

Después de la aplicación del cuestionario a los distintos mandos e involucrados en la organización, se analizan las repuestas de cada uno de los entrevistados que se almacenaron en audio, en cada caso, se organiza la información de cada bloque describiendo las actividades detectadas de acuerdo al bloque del

cuestionario que hace referencia a una área de proceso. Con la información reunida se realiza un diagrama que permite la visualización de cada proceso descrito y detectado.

4.3 Diccionario de términos:

Cuando no existe un proceso formal dentro de la organización y los criterios no se encuentran estandarizados, cada entrevistado maneja su propia terminología, por lo tanto es necesario por cada actor establecer un diccionario de términos que incluya toda el lenguaje que el entrevistado maneja y que servirá para aclarar posibles sesgos al momento de definir los procesos.

4.4 Trazabilidad de las actividades y procesos:

En esta actividad se conjunta cada proceso descrito por cada actor con sus similares, se contrastan las respuestas, elementos del proceso y diagramas para obtener un proceso general, así se establecen las actividades en común que según cada entrevistado se realizan, se logra entonces la consolidación del conocimiento tácito que a cada nivel se tiene dentro de la organización. Con los criterios trazados correctamente, la siguiente actividad consiste en hacer una descripción de los elementos comunes y crear un diagrama del proceso obtenido, basado en las actividades coincidentes, lo que permite la visualización del conocimiento tácito acerca de las tareas que se realizan dentro de la organización. (Véase Fig. 4).

4.4 Mapeo del proceso hacia CMMI-DEV v1.3:

El proceso general y los elementos detectados se mapean hacia una lista de revisión de los elementos que conforman un proceso según CMMI-DEV v1.3, así el proceso obtenido se formaliza y los elementos están listos para ingresarse en la ontología.

5. Aplicación de la herramienta basada en la ontología de CMMI-DEV v1.3 a los procesos de la organización

Cuando se han establecido la situación actual de los procesos y se han identificado las actividades y elementos que conforman dichos procesos mediante el análisis de la extracción del conocimiento tácito de la organización, toda la información se introduce en la herramienta para evaluar la madurez de los procesos respecto a los procesos del modelo CMMI-DEV V1.3. Con la información de los procesos de la organización introducida mediante la interfaz de la herramienta, automáticamente los razonadores implementados comienzan a realizar inferencias, y finalmente muestran en pantalla los procesos que se cumplen, las actividades faltantes y el nivel de madurez en el que se encuentra la organización respecto a CMMI-DEV v1.3. Con estos resultados se identifican posibles hallazgos de mejora para implementar en la organización.

6. Caso de estudio

Para poner en práctica la herramienta ontológica propuesta en este documento, se llevó a cabo un caso de estudio en el Centro de Desarrollo de Software de la Universidad Politécnica de Zacatecas (CDS-UPZ). Tras obtener toda la información de los procesos mediante el método que se describió de manera general en la sección 4, para la extracción del conocimiento tácito (Véase Figura 2), se ejecutó la herramienta ontológica y se cargó la información obtenida, así se corrieron los razonadores y de acuerdo a los procesos que se llevan a cabo la herramienta arrojó el resultado de que la organización aún se encuentra en el nivel 1 o inicial de CMMI-DEV v1.3. Asimismo se detectó con que elementos y actividades no se está cumpliendo, esto permite saber que, aún y cuando se encuentra en el nivel de partida, la

organización puede mejorar sus procesos para ir escalando en la calidad que es requerida en el mercado actual.

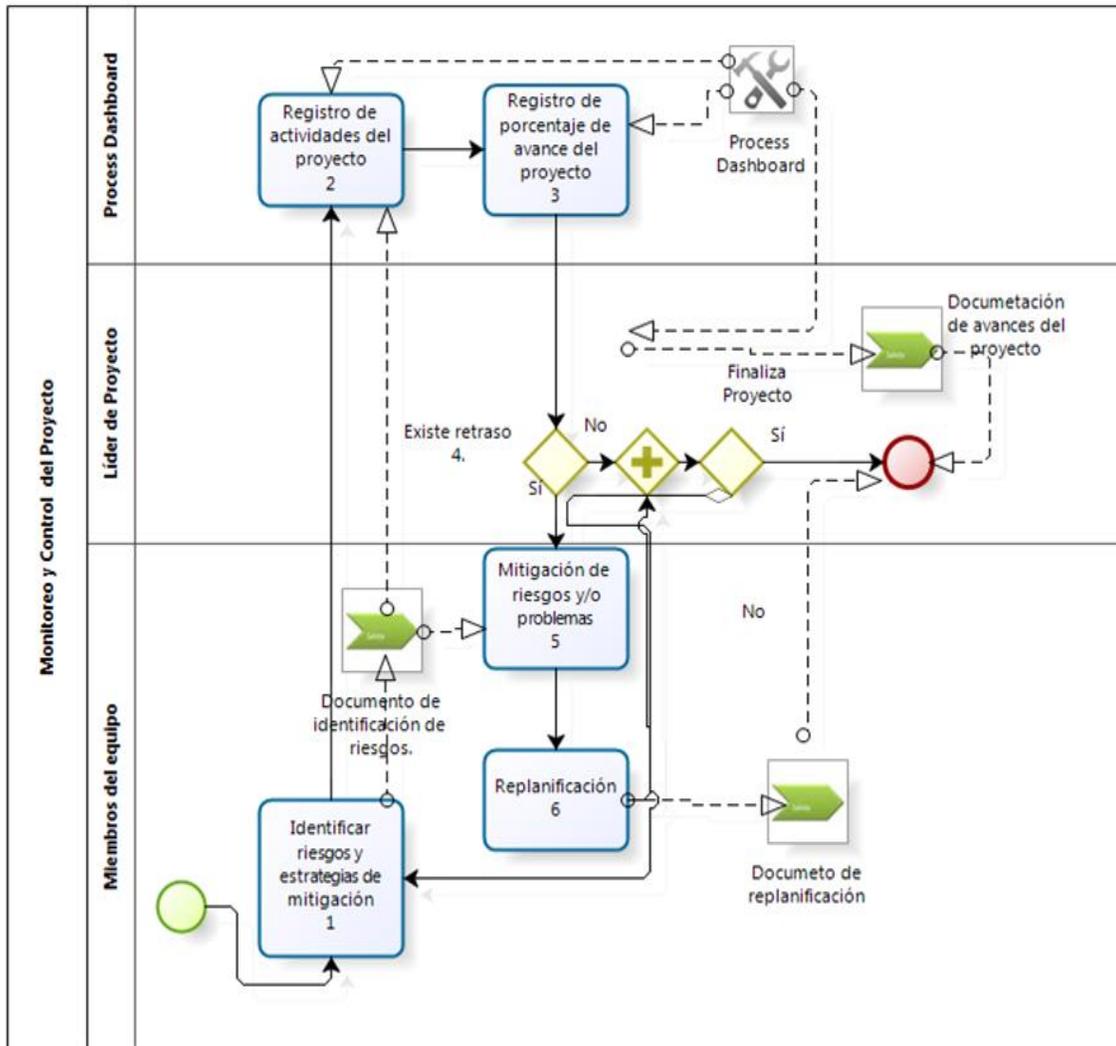


Figura 4. Ejemplo de proceso general obtenido de las actividades del método para la extracción del conocimiento.

Estos hallazgos de mejora son el resultado de evaluar los procesos obtenidos a través del método para la extracción del conocimiento en la herramienta ontológica. La carencia de actividades en las actividades referente al proceso de planificación, formación y a la mayoría de los procesos en los cuales se

compararon los procesos, se detectaron. En resumen los siguientes hallazgos de mejora son:

- No se cuenta con procesos bien establecidos.
- En las planeaciones se sobrestiman las duraciones de los proyectos.
- No se cuentan con roles específicos.
- No existe un plan de formación para los integrantes del equipo de desarrollo.
- No se cuenta con una metodología formal de desarrollo de software.

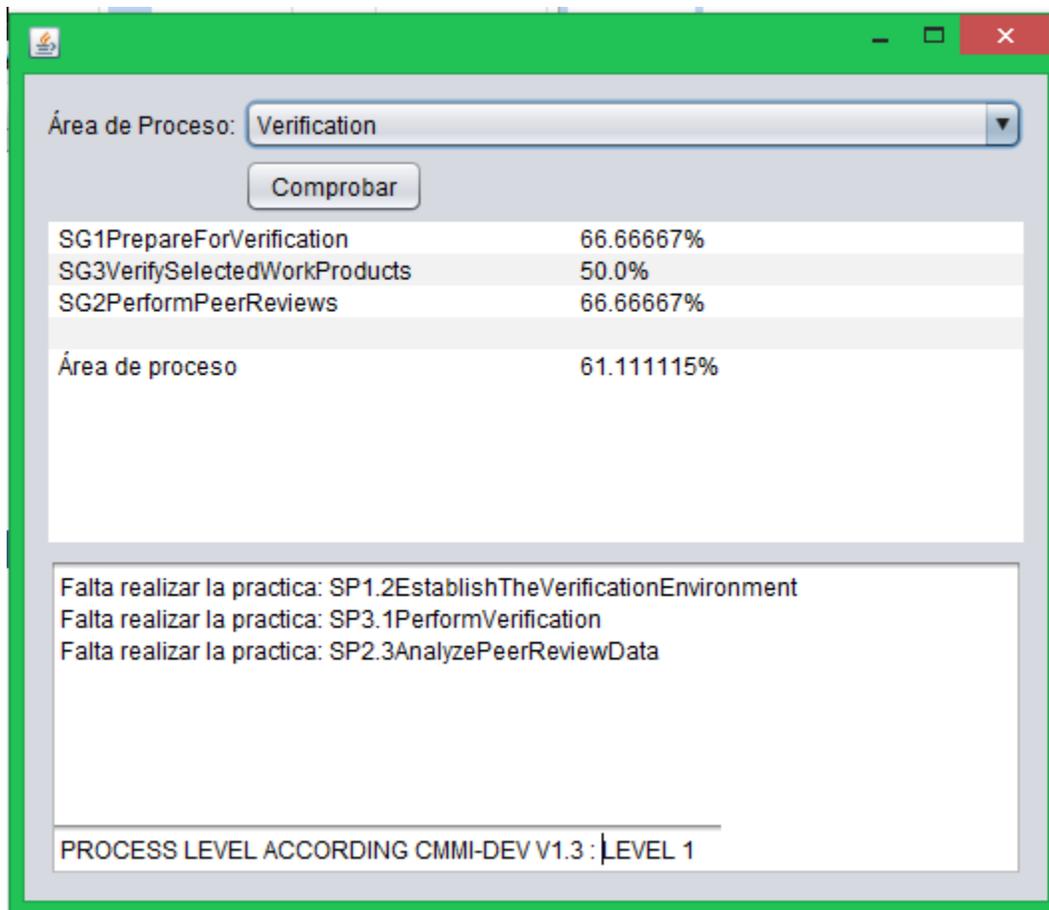


Figura 5. Herramienta de software creada en base a la ontología de CMMI-Dev V1.3

La Fig. 5 muestra como después de introducir las actividades y elementos del proceso general obtenido en el la extracción del conocimiento tácito de la organización, la herramienta mediante los razonadores calcula el porcentaje respecto a las áreas de procesos establecidas en la ontología de CMMI-DEV V1.3 que se realizan en la organización y muestra las actividades faltantes para cumplir con el modelo y el nivel del proceso de acuerdo a CMMI-DEV V1.3., esto se realiza por cada área de proceso y al final se establece una evaluación general de todo el proceso de la organización.

7. Conclusiones y Trabajo Futuro

Mediante la ejecución del método para extraer el conocimiento tácito y con la ayuda de la herramienta ontología basada en el modelo CMMI-DEV v1.3 se logra detectar hallazgos de mejora en las organizaciones como son las pymes. Esta herramienta ontológica se plantea como un soporte de bajo costo para las organizaciones, pues uno de los problemas que enfrentan estos organismos es la falta de presupuestos para identificar hallazgos de mejora con respecto a un modelo o estándar de calidad, por lo que la herramienta presenta una oportunidad para identificarlas. Además, en el caso de estudio al implementar la herramienta permitió detectar que la pyme evaluada estaba en un nivel 1 aún, el nivel inicial de CMMI-DEV v1.3.

Como trabajo futuro se pretende en aplicar las soluciones propuestas en el CDS-UPZ, para observar la efectividad del método completo, y evaluar su funcionamiento si los resultados son positivos, se propone el utilizar la herramienta en otras organizaciones y extender la herramienta para un entorno hacia otros modelos. También se estará desarrollando la automatización del método para la extracción del conocimiento, ya que es una parte principal para el establecimiento de los procesos los cuales se comparan en la herramienta ontológica.

Referencias

Espa, R. (2006). Española de Calidad e Ingeniería del Software.

Halit Soydan, G. (2012). A Partial Formalization of the CMMI-DEV—A Capability Maturity Model for Development. *Journal of Software Engineering and Applications*, 05(10), 777–788. doi:10.4236/jsea.2012.510090

Noy, N. F., & McGuinness, D. L. (2005). Desarrollo de Ontología Para Crear Tu Primera Ontología, 1–29. Report, S. (2003). Demonstrating the Impact and Benefits of CMMI ® : An Update and Preliminary Results, (October).

Santos, G., Montoni, M., Vasconcellos, J., Figueiredo, S., Cabral, R., Cerdeiral, C., ... Rocha, A. R. (2007). Implementing Software Process Improvement Initiatives in Small and Medium-Size Enterprises in Brazil. *6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007)*, 187–198. doi:10.1109/QUATIC.2007.22

Soydan, G. H., & Kokar, M. M. (n.d.). An OWL Ontology for Representing the CMMI-SW Model.

Muñoz Mirna, Mejía Jezreel, “Establishing Multi-model Environments to Improve Organizational Software Processes” Results of the 2013 World Conference on Information Systems and Technologies (WorldCIST'13) held at Algarve, Portugal, March 27-30, 2013.

Notas biográficas:



J. Guadalupe Uribe Dévora es Ingeniero en Sistemas Computacionales, egresado de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés es el desarrollo de software basado en procesos, la arquitectura de software y los modelos de calidad, ha desarrollado varios sistemas de software y sitios web mediante el dominio de varios lenguajes de programación.



Edgar Felipe Uribe Rayas es Ingeniero en Sistemas Computacionales, egresado de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés es el desarrollo de software basado en procesos, modelos de calidad, la seguridad informática y la arquitectura de software, ha desarrollado varios sistemas de software y sitios web mediante el uso de varios lenguajes de programación.



Jorge Alexis Márquez del Real es Ingeniero en Sistemas Computacionales, egresado de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés es la y arquitectura de software, la aplicación de modelos matemáticos a los procesos de software, ha desarrollado varios sistemas de software y sitios web mediante el dominio de varios lenguajes de programación.



Claudia Valtierra Alvarado es una Ingeniero en Sistemas Computacionales, egresada de la Universidad Politécnica de Zacatecas (UPZ), actualmente estudia la Maestría en Ingeniería del Software en el Centro de Investigación en Matemáticas (CIMAT) Unidad Zacatecas. Su interés son los procesos de software, el estudio de la caracterización de organizaciones de desarrollo y el diseño web, ha desarrollado varios sistemas de software y sitios web mediante el dominio de varios lenguajes de programación.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Marcadores para la Realidad Aumentada para fines educativos

María Reina Zarate Nava

Universidad Tecnológica del Centro de Veracruz
Av. Universidad #350 Congregación Dos Caminos,
Cuitláhuac, Veracruz, México.
ma.reina.zarate@gmail.com

Cecilio Francisco Mendoza González

Universidad Tecnológica del Centro de Veracruz
Av. Universidad #350 Congregación Dos Caminos,
Cuitláhuac, Veracruz, México.
cecilio.francisco.mendoza@gmail.com

Honorato Aguilar Galicia

Universidad Tecnológica del Centro de Veracruz
Av. Universidad #350 Congregación Dos Caminos,
Cuitláhuac, Veracruz, México.
aguilargh@hotmail.com

Juan Manuel Padilla Flores

Universidad Tecnológica del Centro de Veracruz
Av. Universidad #350 Congregación Dos Caminos,
Cuitláhuac, Veracruz, México.
jmpf33@hotmail.com

Resumen: La Realidad Aumentada (AR) es considerada por algunos expertos como la tecnología en crecimiento para el 2014, en diferentes sectores: ingeniería, educación, juegos y negocios, siendo este último el más usado en él 2013. Sin embargo en el sector educativo la AR se presenta en algunos proyectos que se han desarrollado para diferentes áreas de las ciencias, como Química, Física y Matemáticas, lo anterior con el objetivo de facilitar el proceso de Enseñanza-Aprendizaje. En el presente artículo se presenta el modelo de patrones utilizado para un Proyecto AR enfocado en Química, el cual presenta un conjunto de patrones que fueron valorados para facilitar el rastreo de los objetos y a su vez el entendimiento de algunos materiales de los laboratorios de Química I.

Palabras clave: realidad aumentada, patrones, marcadores, laboratorios virtuales, Química virtual, proceso de aprendizaje.

Augmented Reality Markers for educational purpose

Abstract: Augmented Reality (AR) is considered by some experts as a growing technology in 2014, in sectors like engineering, education, gaming and business, but business AR has been the most used in 2013. However, in the education sector, AR appear in some projects that have been developed for areas such as science, chemistry, physics and math, all of them with one purpose, to facilitate the learning process. This paper shows a model of patterns used for an AR project focused on chemistry, which presents a set of patterns that were assessed to facilitate tracking of objects and understanding some material of Chemistry lab I subject.

Keywords: augmented reality, patterns, markers, virtual labs, Chemistry virtual, learning process.

1. Introducción

El poder computacional que se tiene actualmente ha permitido desarrollar nuevo software con tecnología de realidad aumentada para apoyo didáctico, el cual busca asegurar que la impartición de los planes de estudio sea completa, así mismo hacer que la transferencia de los conceptos sea de manera clara y asertiva, permitiendo al estudiante desarrollar un conocimiento preciso del concepto del fenómeno observado.

Desde hace una década la programación de Software de realidad aumentada ha tomado gran interés en diferentes sectores tales como: educativo, industrial, servicios y recreativos etc. Los primeros indicios de la realidad aumentada (AR, por sus siglas en inglés) se encuentran en el mundo cinematográfico y en la fotografía (Abud Figueroa, 2012). Se define a la realidad aumentada como la integración de objetos virtuales y se menciona que su nacimiento fue en los años setenta, con inventos como el “Sensorama” y el “head-mounted Display”, los cuales dieron entrada a las primeras publicaciones sobre el tema y la inversión de las compañías en investigación e impulsar el surgimiento de los ahora Smartphone, los cuales actualmente son el soporte más popular para la AR (Arroyo-Vázquez, 2011). Es importante mencionar que la AR no está enfocada a los Smartphone, sino a cualquier dispositivo que tenga integrada una cámara y una pantalla para mostrar el resultado y un software para detectar las referencias y superponer los objetos.

En el campo de la educación, se tienen actualmente distintas aplicaciones de realidad aumentada como: libros aumentados como el “MagicBook” o “El Libro de los Hechizos” (Sony Computer Entertainment Europe, 2013), y en México se ha aplicado en diferentes sectores del gobierno como el del fomento a la lectura con los “LibroPuertos” en la Ciudad de México, D. F. pero nulos en el caso de Química para secundaria. Es importante señalar que actualmente en México no se cuenta con aplicaciones enfocadas al área de Química, pero a nivel internacional existe una empresa que ha trabajado con proyectos aumentados orientados a las ciencias, el proyecto se denomina LearnAR (<http://www.learnar.org/chemistry.html>) y su funcionamiento consiste en mover un marcador que representa un metal y un marcador de solución para que se visualice si existe una reacción en el metal.

En el presente artículo se muestra el desarrollo de marcadores, los cuales son imágenes que una aplicación AR puede reconocer fácilmente (Craig, 2013) y permiten incorporar al entorno real una serie de objetos a través de la cámara; de esta manera se busca generar un laboratorio virtual para realizar prácticas

de Química I con el objetivo de mejorar el proceso enseñanza-aprendizaje e incrementar el nivel académico de los estudiantes de nivel Secundaria, lo cual se verá reflejado en la prueba internacional del Programa Internacional de Evaluación de Estudiantes (PISA, por sus siglas en inglés).

2. Trabajos relacionados

Se han realizado distintos proyectos en el campo de la educación, los cuales buscan la adopción del AR en entornos académicos. Dichas aplicaciones se desarrollan para mejorar la educación y la eficiencia de la formación de los estudiantes (Lee, 2012). Uno de los proyectos orientados al área de la Química es el propuesto por Fjeld y Voegtli denominado “Augmented Chemistry”; éste consiste en un entorno interactivo de trabajo para moléculas en 3D, en donde se muestra a los estudiantes en que consiste un átomo o una molécula a través de la AR (Fjeld, 2002). Los marcadores del anterior proyecto consisten de un patrón sencillo en color blanco y negro, donde no se muestra ninguna relación con la Química, la sencillez del patrón se debe a que en total solo se trabajaron con 6 marcadores que colocaron en un cubo.

Otro de los proyectos es el de la empresa “The Schools Network” de Inglaterra, que contiene un conjunto de aplicaciones en el proyecto “LearnAR” (Specialist Schools and Academies Trust, 2013), en el cual uno de los proyectos que se presenta en el área de Química es el de “Reactividad”, este muestra una serie de metales y su reacción al ser mezclados con una solución. Los marcadores de dicho proyecto contienen un patrón sencillo, el cual consiste de un borde de color y en el centro un cuadro relleno del mismo color. Cada color está relacionado con un metal o solución según corresponda; la poca elaboración en el patrón es debido a que solo trabajan con seis metales y cuatro soluciones. Lo anterior repercute al momento de que un estudiante ve el

marcador, esté no le induce nada relacionado con la Química, que es lo que se busca en el modelo propuesto.

3. Modelo propuesto

Como se mencionó en la introducción, el objetivo del proyecto es permitir a los estudiantes realizar la experimentación en diferentes prácticas de Química I mediante un simulador virtual implementando la AR, en el cual puedan llevar a cabo las prácticas sin necesidad de contar con todo el equipamiento, materiales e instrumental que debe contar un laboratorio de Química completamente equipado.

Para cumplir con lo mencionado anteriormente, el primer paso fue determinar los marcadores “markers”, los cuales son imágenes únicas que servirán como referencia para cada uno de los elementos del laboratorio. Para determinar el diseño del marcador se analizaron las cantidades que se necesitarían de los mismos, para ello se consideraron los ciento dieciocho elementos de la tabla periódica, así como ochenta y seis instrumentos de laboratorio de Química y además material extra que se puede requerir en las diferentes prácticas, por ejemplo, globos, pastillas efervescentes u otros, obteniendo un aproximado de 210 marcadores diferentes.

Los marcadores necesitan un patrón único, el cual le permitirá a la cámara reconocer y determinar el objeto u objetos que debe mostrar. El primer patrón generado con un nivel de una estrella (eficiencia baja) se muestra en la Figura 1, el cual fue descartado porque no permite generar de manera aleatoria cierto número de marcadores, dado que tendía a repetir patrones. Este marcador sólo contiene un patrón sencillo, formado por una cuadrícula y utilizando una escala de grises, también se le incorporó la figura del material con ningún patrón en su interior y en la parte inferior se colocó el nombre del equipo sobre un fondo blanco.



Figura 1. Primer marcador generado para el proyecto de Laboratorio Virtual.

El siguiente patrón mostrado en la Figura 2 es ligeramente diferente en comparación a la Figura 1, este incluye un fondo que cuenta con un patrón propio y se mantuvo la forma del matraz y el nombre completo del mismo. Este patrón fue considerado con un nivel medio de eficiencia, dado que obtuvo 5 estrellas según la evaluación realizada con el software de evaluación "Target Manager" de Vuforia.



Figura 2. Segundo marcador generado para el proyecto de Laboratorio Virtual.

En la Figura 3, se muestra el patrón que actualmente se utiliza en la aplicación, este patrón incluye el logotipo del proyecto, el equipo que se va a utilizar cuyo fondo ya cuenta con un patrón propio, el nombre del equipo y el fondo base se encuentra también con un patrón en escala de grises. Cabe señalar que los colores no influyen en la eficiencia del marcador en la AR.



Figura 3. Marcador que actualmente se utiliza en el proyecto de laboratorio virtual y cuenta con una evaluación de 5 estrellas.

Actualmente se están trabajando los diferentes equipos del laboratorio, pero con el equipo más delgado, se encontró el problema de no alcanzar el nivel deseado dado que el patrón dentro del equipo no logra apreciar, por ejemplo la pipeta. Por tal razón, se está trabajando en una nueva versión de marcador, donde se le incorpora un nuevo patrón para permitir la generación de n marcadores. Como se aprecia en la Figura 4, el nuevo patrón de fondo es más reducido y se incorpora un patrón diferente de lado derecho de la imagen, esto permitirá diferenciarlos sin importar el grosor del equipamiento o el tipo de material.

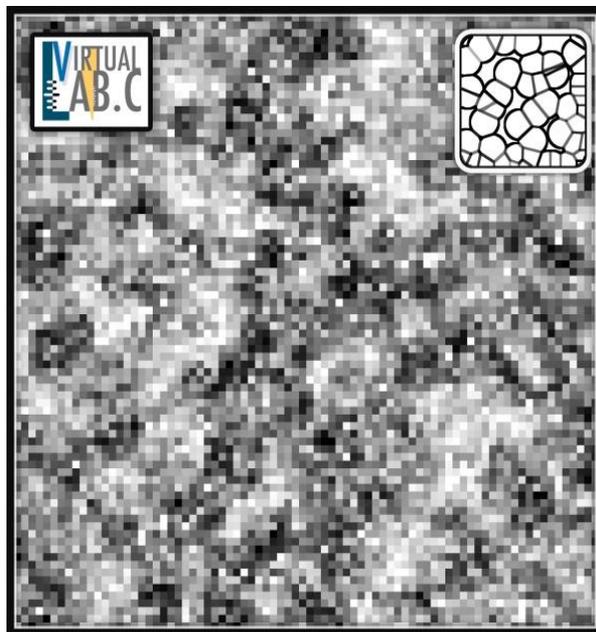


Figura 4. Nuevo marcador, generado por las limitantes que surgen cuando el equipamiento es muy delgado.

Este modelo de patrones en conjunto con el laboratorio virtual, permitirá trabajar con los tres canales de aprendizaje: auditivo, kinestésico y visual. Los alumnos visuales son atraídos por el objeto 3D y el patrón mostrado en los marcadores; los kinestésicos podrán interactuar con cada uno de los objetos y realizar experimentos sin riesgo; los auditivos podrán escuchar un sonido dependiendo de la práctica que esté realizándose.

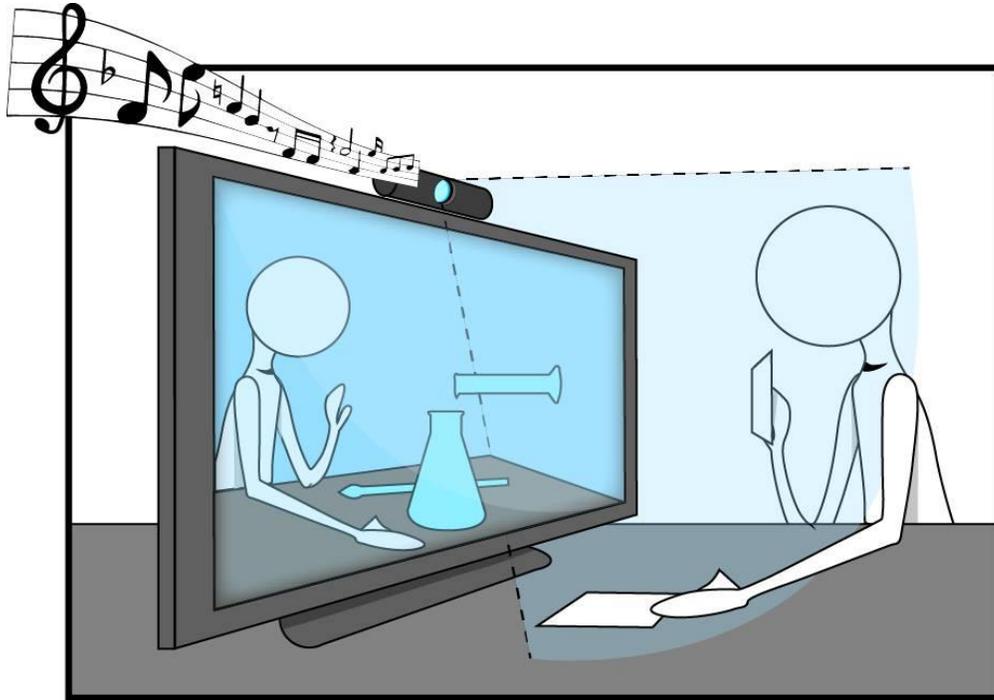


Figura 5. El laboratorio virtual y los canales de aprendizaje.

4. Resultados

Del modelo propuesto de marcadores del laboratorio virtual, ya son interpretados de acuerdo con los patrones establecidos mostrándose el material que corresponde. El tiempo de respuesta para la muestra del objeto tridimensional al momento de enfocar el marcador, varía dependiendo de la ubicación de la cámara, obteniendo tiempos de respuesta de hasta 1 segundo; el anterior resultado se obtuvo instalando la aplicación en un Smartphone con sistema operativo Android 4.1.2. El tamaño preferente del marcador para la interpretación del objeto desde la cámara del Smartphone, es de 17cms x 17cms, aunque un tamaño mínimo para ser detectado por la cámara es de 6cms x 6cms.

En la Figura 6 se muestran los prototipos realizados, los cuales fueron retroalimentados por especialistas del área, quienes expresan la utilidad de los

marcadores incorporando la imagen del material a utilizar, dado que los alumnos lo relacionan con el objeto en la pantalla. El nombre también resulta importante, los especialistas expresan que los alumnos no ubican al material por el nombre, sino por la imagen, trabajando de esta manera con los tres canales de comunicación: kinésico, auditivo y visual.



Figura 6. Prototipo inicial de la detección del matraz con base en el marcador.

En la Figura 7, se muestra el marcador de la pipeta y el objeto aumentado, dicho objeto resulto complicado de trabajar debido a su grosor, el cual no permitía enfocar el patrón ubicando dentro de la pipeta. Como se mencionó anteriormente, dicho problema fue resuelto con la incorporación de nuevos elementos en el marcador.

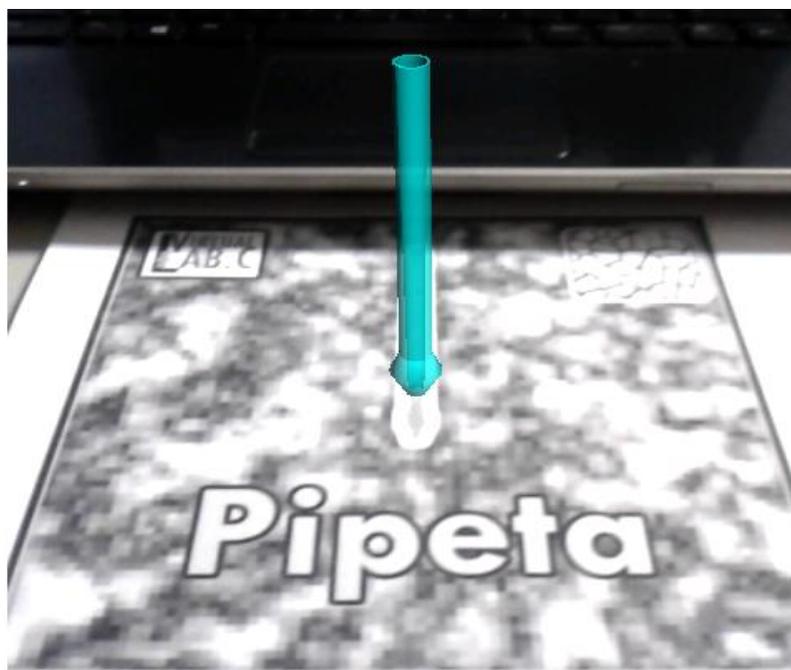


Figura 7. Marcador de la pipeta con patrones más eficientes.

Los objetos anteriores serán mostrados dentro de un laboratorio virtual, dicho escenario permitirá que los alumnos se sientan familiarizados con el ambiente en un laboratorio. Se busca que los materiales como el matraz y la pipeta, se muestren sobre el laboratorio virtual, creando un ambiente aumentado relacionado con la Química.

Los prototipos anteriores, han permitido comprobar la importancia de un patrón eficiente para generación de n marcadores, permitiendo cubrir un total de doscientos diez, requerido por el proyecto. La elaboración de cada uno de los patrones dio como resultado un marcador que es eficiente, completo, y provee buen nivel de rastreo para realizar la diferencia entre objetos, en comparación con otros marcadores, como se muestra en la Tabla 1.

<p>Marcadores del laboratorio virtual de Química.</p>	<p>Marcadores Learn AR en su Proyecto: Chemistry Reactivity. (Specialist Schools and Academies Trust, 2013)</p>
<div data-bbox="355 380 740 779" data-label="Image"> </div> <p data-bbox="293 821 805 993">El marcador ofrece buen nivel de rastreo y el patrón permite generar una gran variedad de marcadores. Al estudiante permite relacionarlo con el material del laboratorio de Química.</p>	<div data-bbox="894 390 1265 779" data-label="Image"> </div> <p data-bbox="833 821 1331 993">En este proyecto el marcador contiene patrones repetitivos y contiene bajo nivel de rastreo. Como tal, no le permite al estudiante relacionarlo con un elemento de Química.</p>

Tabla 1. *Tabla comparativa del marcador propuesto con los marcadores del proyecto Learn AR.*

Cuando el estudiante realice una combinación de marcadores obtendrá un resultado como el mostrado en la Figura 8, el cual dependerá de la calidad seleccionada por el usuario.

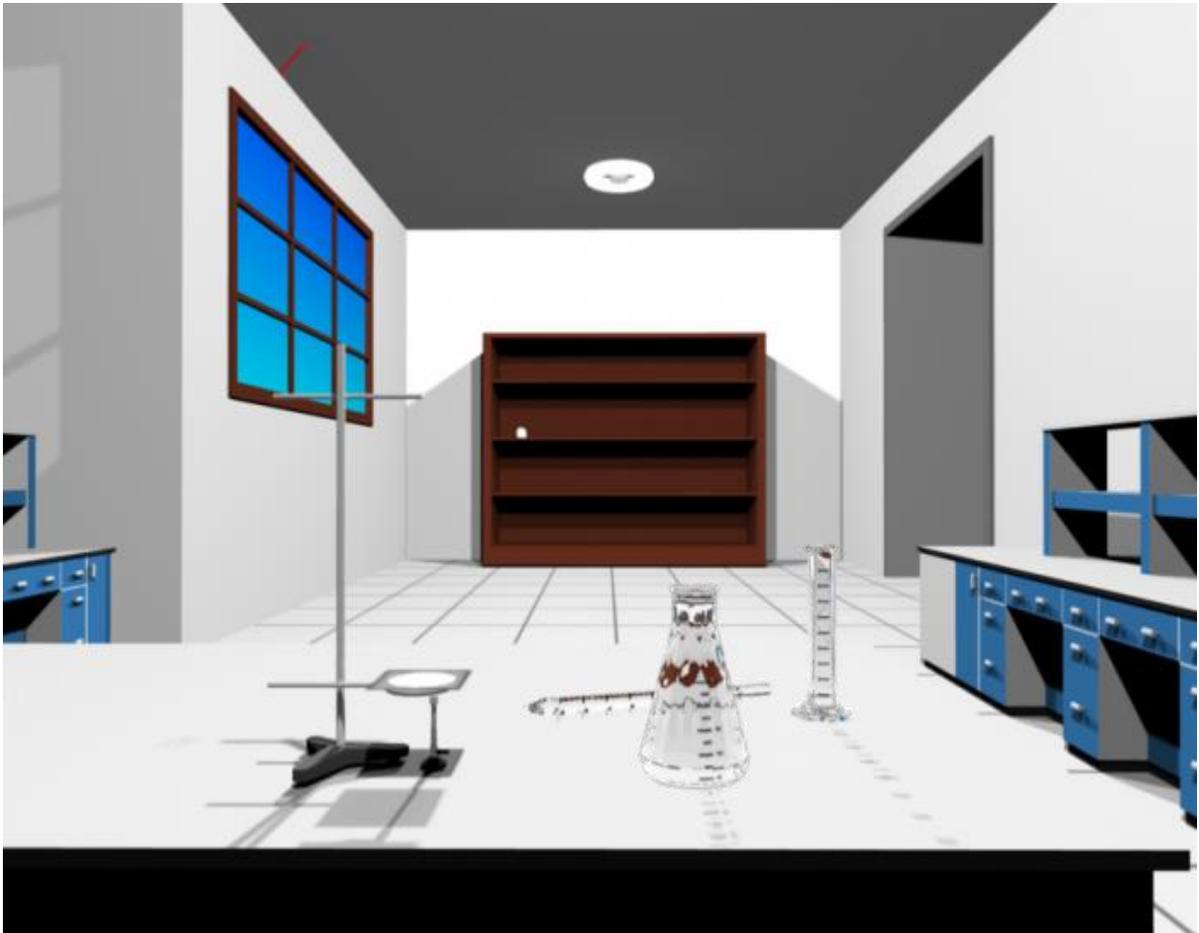


Figura 8. Ambiente aumentado para el laboratorio virtual de Química.

5. Conclusión

El modelo presentado de patrones del laboratorio virtual para el desarrollo de han permitido comprobar la importancia de un patrón eficiente para generación de n cantidad de marcadores, permitiendo cubrir la cantidad requerida por el proyecto. También se ilustraron los patrones y su uso para la representación de los objetos, donde la calidad de los objetos puede ser editada dependiendo de la capacidad del dispositivo en donde se muestren los objetos.

El uso de estos patrones provee un rastreo de objetos específico, minimizando el error al confundir marcadores en la aplicación, y es una alternativa para el desarrollo de aplicaciones AR que impliquen la generación de n marcadores. Como trabajo futuro se contempla implementar “Kinect” para la manipulación de los objetos, para que los estudiantes puedan interactuar con los mismos y hacer el ambiente aumentado más real. Además de que el sonido proporcionado por la aplicación pueda variar dependiendo de las acciones realizadas por el estudiante a través del kinect.

Referencias

- Abud Figueroa, M. A. (2012). Modelo de Objetos de Aprendizaje con Realidad Aumentada. Revista Internacional de la Educación en Ingeniería, 1-7.**
- Arroyo-Vázquez, N. (2011). La realidad aumentada al alcance de todos: creando capas de datos geolocalizados. Anuario ThinkEPI, págs. 269-271.**
- Craig, A. B. (2013). Understanding Augmented Reality Concepts and Applications. USA: Elsevier.**
- Bernat, A., & Catalá, A. (2008). VideoJuegos y Aprendizaje (1 ed.). España: GRAÓ.**
- Fjeld, M. &. (2002). Augmented chemistry: an interactive educational workbench. Proceedings of the international symposium on mixed and augmented reality (ISMAR '02).**
- Lee, K. (2012). Augmented Reality in Education and Training. TechTrends , 13-16.**
- Sony Computer Entertainment Europe. (2013). Wonderbook: EL libro de los Hechizos. Recuperado el 14 de Marzo de 2013, de Sony Computer Entertainment Europe: <http://es.playstation.com/bookofspells/>**
- Specialist Schools and Academies Trust. (2013). <http://www.ssatuk.co.uk/about/>. Recuperado el 10 de 02 de 2013, de <http://www.ssatuk.co.uk/>: http://www.learnar.org/assets/pdf/LAR_resource_combined.pdf**

Notas biográficas



María Reina Zarate Nava estudio la Maestría en Redes y Sistemas Integrados del Laboratorio Nacional de Informática Aplicada y actualmente labora como Profesor-Investigador en la Universidad Tecnológica del Centro de Veracruz. Es programador certificado por parte de Oracle y Microsoft en tecnologías como: ASP.Net, Silverlight y Java; además es mentor certificado por parte de la Universidad Carnegie Mellon. Su interés se encuentra actualmente en la Realidad Aumentada asesorando distintos proyectos que utilicen esta tecnología.



Cecilio Francisco Mendoza González recibió el grado de Master en Animación por parte de la Universidad Pompeu Fabra de Barcelona, los principales intereses radican en la animación de personajes antropomórficos y el desarrollo de historias basadas en el realismo mágico.



Honorato Aguilar Galicia obtuvo el grado de Maestro en Ciencias de la Computación por parte del Instituto Politécnico Nacional en el Centro de Investigación en Computación, con la realización de una tesis titulada “Extracción automática de información semántica basada en estructuras sintácticas”. Ha publicado artículos para el Instituto Politécnico Nacional y su interés actual radica en la investigación de tecnologías de Realidad Aumentada y el desarrollo de nuevos procesos para la norma NMX-I-059/03-NYCE-2005 dentro del Centro de Desarrollo de Software de la Universidad Tecnológica del Centro de Veracruz.



Juan Manuel Padilla Flores La Universidad Autónoma Metropolitana me otorgo el grado de Doctor en Ciencias Química en el año del 2010. Realice tres estancias posdoctorales en la universidad del Papaloapan, en la Universidad Autónoma Metropolitana y la Universidad de Guadalajara respectivamente, he impartido clases a nivel Licenciatura y posgrado desde 1995 a la fecha en la universidad Autónoma Metropolitana, Universidad del Papaloapan y actualmente en la Universidad Tecnológica Del Centro de Veracruz soy miembro activo del SNI nivel 1 cuento con 10 artículos publicados en revistas de impacto con alrededor de 70 citas y ocupo el puesto de jefe de investigadores.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Metodologías para desarrollar software seguro

Carlos Joaquín Brito Abundis

**Universidad Autónoma de Zacatecas
carlosbreeto@gmail.com**

Resumen: La seguridad ha pasado de ser un requerimiento no funcional, que podía implementarse como parte de la calidad del software a un elemento primordial de cualquier aplicación. Los hackers y grupos criminales evolucionan día a día y se han convertido expertos en explotar las vulnerabilidades de las aplicaciones y sitios en internet. Para hacer frente a estas amenazas, es necesaria la implementación de metodologías que contemplen en su proceso de desarrollo de software la eliminación de vulnerabilidades y la inclusión de la seguridad como un elemento básico en la arquitectura de cualquier producto de software. Este trabajo revisa algunas de las metodologías que contemplan la seguridad en su proceso.

Palabras clave: metodologías, desarrollo de software, procesos ágiles, seguridad, vulnerabilidades.

Methodologies for software security development

Abstract: Security has changed from a non-functional requirement, which could be implemented as a part of software quality, to a key element in any software application. Hackers and criminal groups evolve every day and they have become expert in exploiting vulnerabilities in applications and websites. To address these threats, it is necessary that organizations implementing

methodologies that include activities focused on eliminating vulnerabilities and integrating security as a basic element in the software development process. This paper reviews some of the methodologies that provide security activities in the software development process.

Keywords: methodologies, software development, agile processes, security, vulnerabilities.

1. Introducción

La sociedad se encuentra vinculada innegablemente a la tecnología; sin embargo, su mismo uso tan amplio hace que existan grandes riesgos en cada una de las aplicaciones de la tecnología. Además, los riesgos evolucionan y aumentan con la tendencia en el uso de tecnologías como el cloud computing, Web 2.0 y aplicaciones para dispositivos móviles (Laskowski, 2011). En una organización se pierden aproximadamente \$6.6 millones de dólares por cada brecha de seguridad de TI, estas actividades ilícitas tienen ganancias de hasta 114 billones de dólares anualmente y 431 millones de víctimas anuales, es decir, 14 víctimas de cibercrimen cada segundo (Norton, 2012). Un escenario mundial que no puede pasar desapercibido por la gente relacionada al desarrollo de software y empezar a optar por metodologías que garanticen el lugar que le corresponde a la seguridad.

El objetivo de este trabajo es presentar dos de las metodologías de desarrollo que permiten producir productos de software seguros. Después se presenta un análisis de las metodologías que permiten producir productos de software seguros, una comparación entre ellas y por último las conclusiones y los trabajos futuros que se desprenden de éste trabajo.

2. Metodologías de desarrollo tradicionales

La clave de un software seguro, es el proceso de desarrollo utilizado. En el proceso, es donde se produce el producto que pueda resistir o sostenerse ataques ya anticipados, y recuperarse rápidamente y mitigar el daño causado por los ataques que no pueden ser eliminados o resistidos. Muchos de los defectos relacionados con la seguridad en software se pueden evitar si los desarrolladores estuvieran mejor equipados para reconocer las implicaciones de su diseño y de las posibilidades de implementación.

La manera en que se desarrolla software ha evolucionado, desde la forma de “code and fix” (codificar y después arreglar), en la que un equipo de desarrollo tiene la idea general de lo que quiere desarrollar, pasando a metodologías formales (RUP, Proceso Unificado, PSP/TSP, entre otras) en las que existe una planeación detallada del desarrollo, desde la elicitación, documentación, requerimientos, diseño de alto nivel y la inspección (Vanfosson, 2006). Los métodos ágiles permiten tener un desarrollo iterativo, con ciclos de entrega continuos, un contacto con el cliente permanente; permitiendo que se incluyan como parte de los stakeholders a los administradores de riesgo de la empresa, certificadores y al personal responsable de las políticas de seguridad. Los métodos ágiles se rigen bajo el manifiesto de los procesos ágiles, que menciona que la prioridad más alta es la satisfacción del cliente a través de entregas tempranas y continuas de un producto con valor (Fowler & Highsmith, 2001).

Sin embargo; las metodologías mencionadas, tienden enfocarse en mejorar la calidad en el software, reducir el número de defectos y cumplir con la funcionalidad especificada (Davis, 2005); pero en la actualidad, también es necesario entregar un producto que garantice tener cierto nivel de seguridad. Hay metodologías que contemplan durante su proceso, un conjunto de actividades específicas para remover vulnerabilidades detectadas en el diseño

o en el código, la aplicación de pruebas que aportan datos para la evaluación del estado de seguridad, entre otras actividades relacionadas para mejorar la seguridad del software.

3. Metodologías enfocadas al desarrollo de software seguro

Existen varias metodologías que establecen una serie de pasos en búsqueda de un software más seguro y capaz de resistir ataques. Entre ellas se encuentran Correctness by Construction (CbyC), Security Development Lifecycle (SDL), Cigital Touchpoints, Common Criteria, Comprehensive, Lightweight Application Security Process (CLASP), TSP-Secure. El presente artículo estudiará las características de las dos primeras, detallando las fases que las conforman, destacando sus particularidades y al final se realiza una comparativa de ambas.

3.1 Correctness by Construction (CbyC)

Es un método efectivo para desarrollar software que demanda un nivel de seguridad crítico y que además sea demostrable. La empresa Praxis ha utilizado CbyC desde el año 2001 y ha producido software industrial con tasa de defectos por debajo de los 0.05 defectos por cada 1000 líneas de código, y con una productividad de 30 líneas de código por persona al día.

Las metas principales de ésta metodología son obtener una tasa de defectos al mínimo y un alta resiliencia al cambio; los cuales se logran debido a dos principios fundamentales: que sea muy difícil introducir errores y asegurarse que los errores sean removidos tan pronto hayan sido inyectados. CbyC busca producir un producto que desde el inicio sea correcto, con requerimientos rigurosos de seguridad, con definición muy detallada del comportamiento del sistema y un diseño sólido y verificable (Croxford & Chapman, 2005).

3.1.1 Fases de la Metodología CbyC

CbyC combina los métodos formales con el desarrollo ágil; utiliza notaciones precisas y un desarrollo incremental que permite mostrar avances para recibir retroalimentación y valoración del producto. La Figura 1, muestra las fases propuestas por ésta metodología para el desarrollo de software.

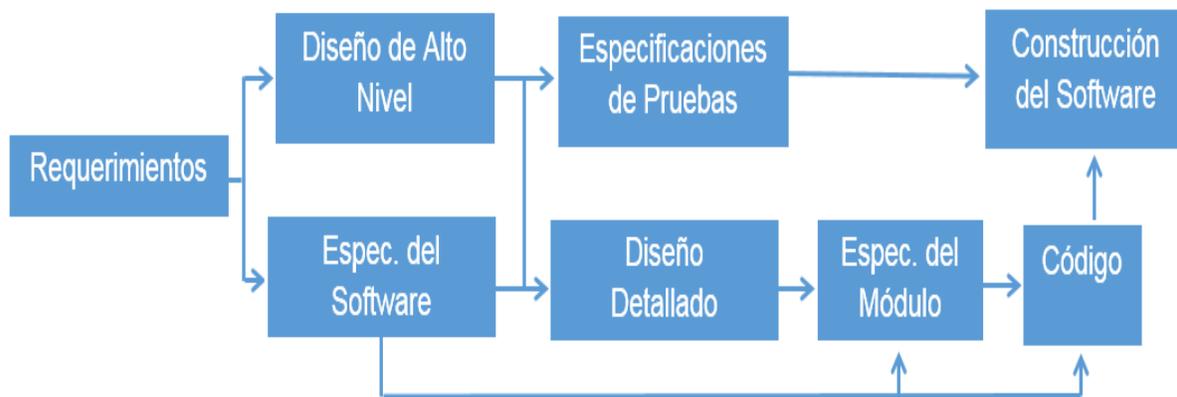


Figura 1. Proceso de desarrollo propuesto por el CbyC (Amey, 2006).

3.1.1.1 Fase de Requerimientos

En la fase de requerimientos se especifica el propósito, las funciones y los requerimientos no funcionales. Se escriben los requerimientos de usuario con sus respectivos diagramas contextuales, diagramas de clase y definiciones operativas. Cada requerimiento debe pasar por un proceso de trazabilidad, así como requerimiento de seguridad a la amenaza correspondiente.

3.1.1.2 Fase de Diseño de Alto Nivel

Se describe la estructura interna del sistema, (distribución de la funcionalidad, estructura de las bases de datos, mecanismos para las transacciones y comunicaciones) la manera en que los componentes colaboran y especifican con mayor énfasis los requerimientos no funcionales de protección y seguridad. Se utilizan los métodos formales para definir el diseño de alto nivel y obtener

una descripción, un comportamiento y un modelo preciso. Los métodos formales han probado ser exitosos para especificar y probar el nivel de 'correctness' y en la consistencia interna de las especificaciones de funciones de seguridad (Jarzombek & Goertzel, 2006).

3.1.1.3 Fase de Especificación del Software

En ésta fase se documentan las especificaciones de la interfaz de usuario (se define el "look and feel" del sistema), las especificaciones formales de los niveles superiores y se desarrolla un prototipo para su validación.

3.1.1.4 Fase de Diseño Detallado

Define el conjunto de módulos y procesos y la funcionalidad respectiva. Se utiliza notación formal para eliminar ambigüedad en la interpretación del diseño.

3.1.1.5 Fase de Especificación de los Módulos

Se define el estado y el comportamiento encapsulado en cada módulo del software tomando en cuenta el flujo de información. Los módulos deberán de tener el enfoque de bajo acoplamiento y alta cohesión; así los efectos serían mínimos en caso de producirse una falla en el flujo de información.

3.1.1.6 Fase Codificación

El concepto de evitar cualquier ambigüedad posible, también se aplica en el código; por ello se sugiere la utilización de SPARK (lenguaje formalmente definido y matemáticamente comprobable, basado en el lenguaje de programación Ada, utilizado en sistema de aeronáutica, sistemas médicos y control de procesos en plantas nucleares) en las partes críticas del sistema por estar diseñado para tener una semántica libre de ambigüedades, permitir realizar análisis estático y ser sujeto de una verificación formal (Brito, 2010). Se

conducen pruebas de análisis estático al código para eliminar algunos tipos de errores y en caso de ser necesario, se realizará una revisión al código.

3.1.1.7 Fase de las Especificaciones de las Pruebas

Para obtener las especificaciones de las pruebas, se toman en cuenta las especificaciones del software, los requerimientos y el diseño de alto nivel. Se efectúan pruebas de valores límites, pruebas de comportamiento y pruebas para los requerimientos no funcionales; pero CbyC no usa pruebas de caja blanca ni pruebas de unidad; todas las pruebas son a nivel de sistema y orientado a las especificaciones.

3.1.1.8 Fase de Construcción del Software

CbyC utiliza el desarrollo de tipo ágil; en la primera entrega, se tiene el esqueleto completo del sistema con todas las interfaces y mecanismos de comunicación; con una funcionalidad muy limitada que se irá incrementado en cada iteración del ciclo.

CbyC es compatible con los principios de los procesos de Personal Software Process/Team Software Process (PSP/TSP) y al combinar resulta en una taza reducción en la taza de defectos. El enfoque técnico de CbyC complementa a PSP para aumentar la capacidad de remoción de defectos en etapas tempranas del desarrollo. (Croxford & Chapman, 2005) . La Figura 2, muestra las tasas de defectos en los cinco niveles del CMM y la tasa cercana cero perteneciente a la metodología CbyC, dejando en claro el beneficio de la implementación de ésta un proceso de desarrollo.

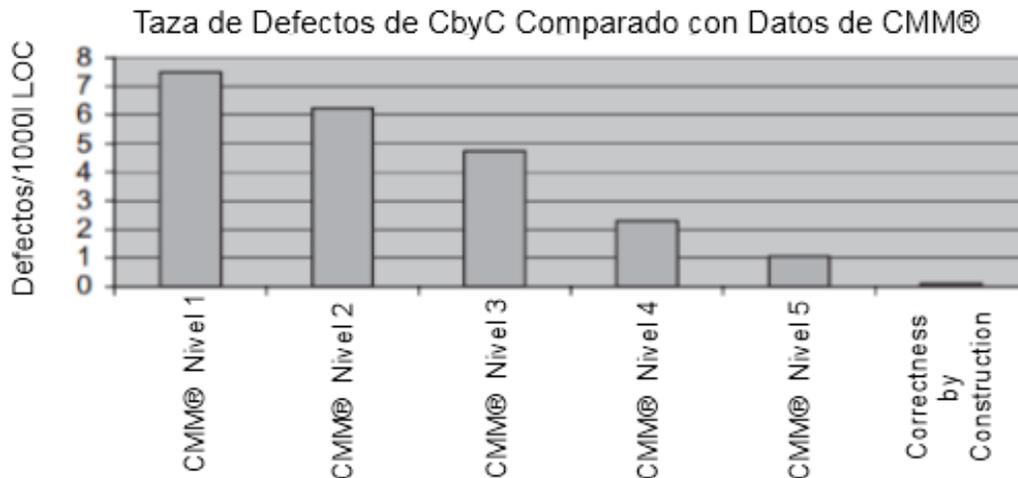


Figura 2. Comparativa de tasa de defectos CMMI contra CbyC (Hall, 2007).

3.2 Security Development Lifecycle (SDL)

Es un proceso para mejorar la seguridad de software propuesto por la compañía de Microsoft en el año 2004; con dieciséis actividades enfocadas a mejorar la seguridad del desarrollo de un producto de software (Peterson, 2011).

Las prácticas que propone SDL van desde una etapa de entrenamiento sobre temas de seguridad, pasando por análisis estático, análisis dinámico, fuzz testing del código hasta tener plan de respuesta a incidentes. Una de las características principales de SDL es el modelado de amenazas que sirve a los desarrolladores para encontrar partes del código, donde probablemente exista vulnerabilidades o sean objeto de ataques (Korkeala, 2011).

3.2.1 Fases de la Metodología SDL

Existen dos versiones del SDL, la versión rígida y la orientada al desarrollo ágil. Las diferencias versan en que la segunda desarrolla el producto de manera incremental y en la frecuencia de la ejecución de las actividades para el aseguramiento de la seguridad. La versión rígida del SDL es más apropiada para equipos de desarrollo y proyectos más grandes y no sean susceptibles a

cambios durante el proceso. SDL ágil es recomendable para desarrollos de aplicaciones web o basados en la web (Wood & Knox, 2012). En la Figura 3, se encuentra el flujo de las fases en la metodología de SDL cabe resaltar la existencia de una etapa previa a los requerimientos, enfocada al entrenamiento en seguridad y la última etapa del proceso que se encarga de darle seguimiento al producto en caso de algún incidente de seguridad.

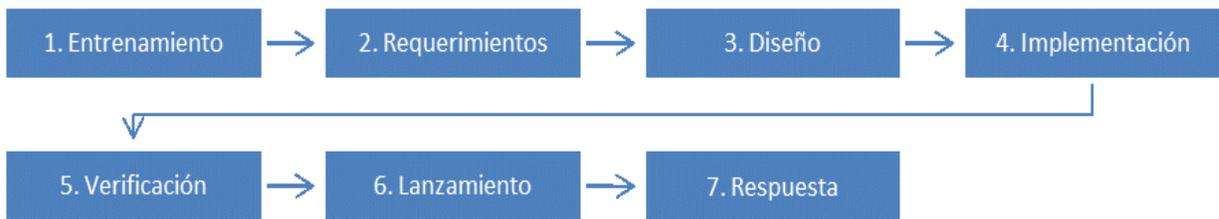


Figura 3. Proceso del desarrollo software del SDL (Microsoft Corporation, 2007).

Las dieciséis actividades de SDL, que consolidan la seguridad, se detallan en la Tabla 1.

Actividades del SDL para la Seguridad	
1. Entrenamiento	5. Verificación
Entrenamiento de seguridad básica	Análisis dinámico
2. Requerimientos	Fuzz Testing
Establecer requerimientos de seguridad	Revisión de la superficie de ataques
Crear umbrales de calidad y límites de errores	6. Lanzamiento
Evaluación de los riesgos de seguridad y privacidad	Plan de respuesta a incidentes
3. Diseño	Revisión de seguridad final
Establecer requerimientos de diseño	Aprobar y archivar lanzamiento
Análisis de la superficie de ataques	7. Respuesta
Modelado de amenazas	Ejecutar el plan de respuesta a incidentes
4. Implementación	
Utilizar herramientas aprobadas	
Prohibir funciones no seguras	
Análisis estático	

Tabla 1. Actividades del SDL para el aseguramiento de la seguridad (Microsoft Corporation, 2007)

3.2.1.1 Fase de Entrenamiento

Todos los miembros de un equipo de desarrollo de software deben recibir una formación apropiada con el fin de mantenerse al corriente de los conceptos básicos y últimas tendencias en el ámbito de la seguridad y privacidad. Las personas con roles técnicos (desarrolladores, evaluadores y administradores de programas) que están directamente implicadas en el desarrollo de programas de software deben asistir como mínimo una vez al año a una clase de formación en materia de seguridad en conceptos fundamentales como: defensa en profundidad, principio de privilegios mínimos, modelos de riesgos, saturaciones de búfer, inyección de código SQL, criptografía débil, evaluación de riesgos y procedimientos de desarrollo de privacidad.

3.2.1.2 Fase de Requerimientos

En la fase de requerimientos, el equipo de producción es asistido por un consultor de seguridad para revisar los planes, hacer recomendaciones para cumplir con las metas de seguridad de acuerdo al tamaño del proyecto, complejidad y riesgo. El equipo de producción identifica como será integrada la seguridad en el proceso de desarrollo, identifica los objetivos clave de seguridad, la manera que se integrará el software en conjunto y verificarán que ningún requerimiento pase desapercibido.

3.2.1.3 Fase de Diseño

Se identifican los requerimientos y la estructura del software. Se define una arquitectura segura y guías de diseño que identifiquen los componentes críticos para la seguridad aplicando el enfoque de privilegios mínimos y la reducción del área de ataques. Durante el diseño, el equipo de producción conduce un modelado de amenazas a un nivel de componente por componente, detectando los activos que son administrados por el software y las interfaces por las cuales se pueden acceder a esos activos. El proceso del modelado identifica las amenazas que potencialmente podrían causar daño a algún activo y establece la probabilidad de ocurrencia y se fijan medidas para mitigar el riesgo.

3.2.1.4 Fase de Implementación

Durante la fase de implementación, se codifica, prueba e integra el software. Los resultados del modelado de amenazas sirven de guía a los desarrolladores para generar el código que mitigue las amenazas de alta prioridad. La codificación es regida por estándares, para evitar la inyección de fallas que conlleven a vulnerabilidades de seguridad. Las pruebas se centran en detectar vulnerabilidades y se aplican técnicas de fuzz testing y el análisis de código estático por medio de herramientas de escaneo, desarrolladas por Microsoft. Antes de pasar a la siguiente fase, se hace una revisión del código en

búsqueda de posibles vulnerabilidades no detectadas por las herramientas de escaneo.

3.2.1.4 Fase Verificación

En ésta parte el software, ya es funcional en su totalidad y se encuentra en fase beta de prueba. La seguridad es sujeta a un “empujón de seguridad”; que se refiere a una revisión más exhaustiva del código y a ejecutar pruebas en parte del software que ha sido identificado como parte de la superficie de ataque.

3.2.1.5 Fase de Lanzamiento

En el lanzamiento, el software es sujeto a una revisión de seguridad final, durante un periodo de dos a seis meses previos a la entrega con el objetivo de conocer el nivel de seguridad del producto y la probabilidad de soportar ataques, ya estando liberado al cliente. En caso de encontrar vulnerabilidades que pongan en riesgo la seguridad, se regresa a la fase previa para enmendar esas fallas.

3.2.1.6 Fase de Respuesta

Sin importar la cantidad de revisiones que se haga al código o las pruebas en búsqueda de vulnerabilidades, no es posible entregar un software cien por ciento seguro; así que, se debe de estar preparado para responder a incidentes de seguridad y a aprender de los errores cometidos para evitarlos en proyectos futuros (Lipner, 2004).

La compañía Microsoft utilizó el proceso de SDL para hacer el Windows Vista, reduciendo en un 50% las vulnerabilidades comparadas con Windows XP SP2. Después lanzó Microsoft Office 2007, con una reducción del 90% de las vulnerabilidades, a causa de dos requerimientos de SDL: del requerimiento de integer overflow y el requerimiento de fuzzing (Microsoft Corporation, 2007).

4. Comparativa de las metodologías CbyC y SDL

Las metodologías presentadas en éste artículo, tienen como objetivo establecer una forma de desarrollar software que sea más seguro y capaz de soportar ataques maliciosos. En la Tabla 2 se hace una comparativa entre las características de ambas.

Atributo/Metodología	CbyC	SDL
Utilización de métodos formales	X	-
Utiliza desarrollo iterativo	X	-
Entrenamiento en temas de seguridad	-	X
Establece requerimientos de seguridad	X	X
Estudia la trazabilidad de los requerimientos	X	-
Asistencia de personal especializado en la seguridad	-	X
Realiza modelado de amenazas	-	X
Análisis estático	X	X
Análisis dinámico	-	X
Fuzz testing	-	X
Revisión de código	X	X
Desarrolla un plan en caso de incidentes de seguridad	-	X

Tabla 2. Comparativa entre las metodologías CbyC y SDL

De la tabla anterior se podría llegar a pensar que SDL representa mejor seguridad en el producto final. Sin embargo CbyC enfoca su proceso en crear un producto que sea correcto y con el menor número de defectos posibles. SDL tiene varias actividades durante todo el proceso para educar, para descubrir vulnerabilidades en el código y para reducir al área de ataque maliciosos. Las dos tienen limitaciones respecto a los lenguajes de programación, ya que por una parte CbyC se debe comprar la versión del lenguaje SPARK que soporta exigencias críticas de seguridad; y por otro lado SDL siendo impulsada por Microsoft utiliza lenguajes y suite de programación familiares a la compañía, al igual que las herramientas proporcionadas únicamente corren bajo ambiente del sistema operativo Windows.

5. Conclusiones

La seguridad ha dejado de ser un atributo de calidad, se encuentra en cada capa de la arquitectura de software, y por lo tanto no se puede dejar como un elemento aislado, sino que es transversal y multidimensional. Los hackers parece que siempre están dos pasos más delante de las organizaciones, si se continúa haciendo software de la manera tradicional; esa brecha será aprovechada y seguirán explotando vulnerabilidades que pudieron haber sido evitadas utilizando metodologías como las que en éste documento se citaron.

Este trabajo presentó de manera general, procesos desarrollo de software que tienen actividades enfocadas a la mejora de la seguridad. La utilización de una metodología de éstas características es imperativa para mitigar y tratar de evitar los ataques que día a día, las organizaciones y las personas son víctimas, causando pérdidas millonarias. Recordando que la mejor metodología es aquella que se adapta al contexto del producto a desarrollar, eso da la pauta para escoger la metodología idónea. En un estudio posterior, se podría formular la integración de la versión ágil del SDL con Scrum y aplicarlo en un caso práctico.

Referencias

Amey, P. (2006). Correctness by Construction. Consultado el 29 de septiembre del 2013, en <https://buildsecurityin.us-cert.gov/articles/knowledge/sdlc-process/correctness-by-construction>

Brito, E. (2010). A (Very) Short Introduction to SPARK : Language , Toolset , Projects , Formal Methods & Certification (pp. 479–490). Portugal: INForum 2010 - II Simpósio de Informática.

Croxford, M., & Chapman, R. (2005). Correctness by Construction : A Manifesto for High-Integrity Software. The Journal of Defense Software Engineering, 18(12), 5–8.

Davis, N. (2005). Secure Software Development Life Cycle Processes : A Technology Scouting Report (pp. 14–20).

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. Consultado el 07 de julio del 2013, en <http://www.pmp-projects.org/Agile-Manifesto.pdf>

Hall, A. (2007). Realising the Benefits of Formal Methods. Journal of Universal Computer Science, 13(5), 669–678.

Korkeala, M. (2011). Integrating SDL for Agile in an ongoing software development project. Cloud Software Finland, 1–17.

Laskowski, J. (2011). Agile IT Security Implementation Methodology (primera ed., pp. 13–21). Birmingham, Reino Unido: Packt Publishing Ltd.

Lipner, S. (2004). The Trustworthy Computing Security Development Lifecycle. Annual Computer Security Applications Conference, pp. 2 – 11.

Microsoft Corporation. (2007). The Trustworthy Computing Security Development Lifecycle The Microsoft SDL Team. Consultado en <http://www.microsoft.com/en-us/download/details.aspx?id=12379>

Norton. (2012). Norton Cybercrime Report 2012 (pp. 1–9). Consultado en <http://us.norton.com/cybercrimereport>

Peterson, G. (2011). Security Architecture Blueprint. Dublin: Secure Application Development. Consultado en <http://secappdev.org/handouts/2011/GunnarPeterson/ArctecSecurityArchitectureBlueprint.pdf>

Vanfosson, T. (2006). Plan-driven vs . Agile Software Engineering and Documentation : A Comparison from the Perspectives of both Developer and Consumer Submitted for the PhD Qualifying Examination. CiteSeerX

Wood, C., & Knox, G. (2012). Guidelines for Agile Security Requirements Engineering. Software Requirements Engineering (pp. 1 –5). Rochester, Nueva York.

Notas biográficas



Carlos Joaquín Brito Abundis tiene el grado de Licenciado en Derecho por la Universidad Autónoma de Durango, actualmente cursa el último semestre de la carrera de Ingeniería en Software en la Universidad Autónoma de Zacatecas. Participó con el proyecto de investigación “Evaluación de la Seguridad de la Red Universitaria mediante Pruebas de Penetración” en el Primer Foro de Jóvenes Investigadores realizado en la ciudad de Fresnillo, Zacatecas. Asimismo, elaboró el poster “Enfoques de Desarrollo para Software Seguro” en el marco del Segundo Congreso Internacional de Mejora de Procesos de Software. Los temas de su interés son pruebas de penetración en dispositivos móviles y la evaluación de la seguridad en las empresas.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Herramienta de gestión de Modelos Ontológicos aplicados a la Mejora de Procesos Software

José Eduardo Guadalupe Gaytán Solís

**Centro de Investigación en Matemáticas, Unidad Zacatecas
Av. Universidad 222, Zacatecas, México
jose.gaytan@cimat.mx**

Ricardo González Saldívar

**Centro de Investigación en Matemáticas, Unidad Zacatecas
Av. Universidad 222, Zacatecas, México
ricardo.gonzalez@cimat.mx**

Edrisi Muñoz Mata

**Centro de Investigación en Matemáticas, Unidad Zacatecas
Av. Universidad 222, Zacatecas, México
emunoz@cimat.mx**

Resumen: Actualmente el software representa un componente principal en el desarrollo de las actividades dentro de las organizaciones. Este hecho hace que la calidad del software sea uno de los principales factores claves a los que se dedica mucho esfuerzo. La calidad del software permite de manera implícita la creación de valor agregado al software durante su desarrollo, resultando en un factor de competitividad en estas organizaciones. Éste trabajo propone una solución inteligente mediante el uso de gestión del conocimiento, con el fin de hacer que la aplicación de teorías de ingeniería de software, modelos y estándares de mejora de procesos sea de fácil implementación y a un menor costo. Específicamente, en este trabajo se presenta el desarrollo de un modelo ontológico basado el Cuerpo de Conocimientos de la Ingeniería de Software (SWEBOK por las siglas en inglés de Software Engineering Body of Knowledge) así como su aplicación práctica mediante un sistema software. Finalmente se presentan los resultados de la implementación de técnicas, métricas y actividades correspondientes al área de conocimiento de pruebas de software de SWEBOK. Así mismo se presentan resultados de la integración del marco base de datos de lenguaje ontológico web (OWLDB por las siglas en inglés ontology web language data base), con el objetivo de almacenar datos de manera persistente durante la aplicación.

Palabras clave: Ontología, Cuerpo de Conocimientos de la Ingeniería de Software, SWEBOK, OWLDB, Bases de Datos, Mejora de procesos de software.

Ontologic Models management tool applied to Software Process Improvement

Abstract: Nowadays, software is a major component in the development of activities within organizations. This fact makes software quality is one of the main key factors which demands much effort. In this way, software quality allows the creation of added value to the software during development, resulting in a competitive factor for the organization. This paper proposes an intelligent solution for a lower cost implementation of software engineering theories, models and process improvement standards by the use of knowledge management. Specifically, this paper presents the development of an ontological model based on the Body of Knowledge Software Engineering (SWEBOK) and its practical application through a software system. Finally this work presents the results of the implementation of techniques, metrics and activities for the area of software testing knowledge of SWEBOK. It also presents results of the ontology web language data base (OWLDB) frame integration, aiming to store persistent data in the application.

Keywords: Ontology, Software Engineering Body of Knowledge, SWEBOK, OWLDB, Data Bases, Software Process Improvement.

1. Introducción

En la actualidad existen numerosos modelos y estándares dentro de la industria que pretenden regular y asegurar la calidad como por ejemplo CMMI, Six Sigma y SWEBOK (Pyzdek, 2003; Abran, Bourque, Dupuis, Moore & Tripp, 2004; CMMI Product Team, 2010). Estas guías metodológicas plantean procesos que establecen una manera estructurada y orientada para mantener una eficiente preparación, operación y mantenimiento de los productos que generan. Por otra parte, los datos significativos para la operación de las empresas son una fuente de conocimiento necesario para el desarrollo de las mismas y por su importancia es necesario que se le dé un trato eficiente de conservación, reutilización y gestión para mantener competitividad. Una manera de gestionar este conocimiento puede generarse a través de la utilización de los modelos ontológicos. Una ontología es una especificación formal de la conceptualización de un área o dominio en particular. Mediante el desarrollo de una ontología se establece un modelo formal, que proporciona heterogeneidad semántica. Este tipo de modelos proveen una comprensión compartida que es procesable por humanos y por aplicaciones informáticas (Cullot, Ghawi & Yétongnon, 2007; Quintanilla, 2005; Barchini & Herrera, 2010; Romá, 2009).

Por otra parte una nueva tendencia en cuanto a la administración del conocimiento es la utilización de un sistema de almacenamiento persistente a través de bases de datos. Este tipo de administración de datos aporta una extensibilidad y amplía el campo de aplicación de los modelos ontológicos (Cullot, Ghawi & Yétongnon, 2007; Trinkunas & Vasilecas, 2007; Aksoy, Alparslan, Bozdağ, Çulhacı, 2011; Heymans et al., 2008). Aún más, mediante la integración de este almacenamiento y los modelos semánticos se crean nuevos modelos y estructuras de información que dan como resultado las conocidas bases de datos inteligentes que explotan “conocimiento” y una vista semántica a la información.

El presente trabajo presenta la elaboración de una ontología basada en guía de cuerpo de conocimientos de la ingeniería de software (SWEBOK por las siglas en inglés de Software Engineering Body of Knowledge) desarrollado por el Instituto de Ingeniería de Software (Abran, Bourque, Dupuis, Moore & Tripp, 2004). A continuación se describe de manera breve el contenido de este trabajo. La sección 2 muestra la propuesta y a la audiencia a la que está destinado el trabajo. En la sección 3 se da a conocer la metodología de desarrollo de la ontología y de la herramienta de software para la administración de la ontología. La sección 4 describe de manera breve el proceso de desarrollo de la ontología y del sistema y, de manera general, el contenido de dicha ontología, además del desarrollo de la herramienta y la implementación con OWLDB . En la sección 5 se muestran las conclusiones del trabajo realizado. Finalmente la sección 6 presenta el trabajo futuro basado en las conclusiones obtenidas.

2. Propuesta

Los modelos de calidad existentes en la actualidad están formados por una gran cantidad de conceptos, guías específicas y reglas que deben seguirse para poder ser implementados (Pyzdek, 2003; Abran, Bourque, Dupuis, Moore & Tripp, 2004; CMMI Product Team, 2010). El manejo de esta información, por parte de las empresas, puede representar una inversión relativamente considerable de recursos que pudiera resultar en un impedimento o retraso en la puesta en marcha de determinado modelo de calidad (Muñoz, Mejía, Muñoz, 2013). Esta dificultad puede llegar a ser manejable de una manera menos costosa a través del uso de la propuesta descrita en este documento. Para ello, se divide este trabajo, en dos partes principales de acción:

1. Desarrollo de la Ontología de SWEBOK. Se trata de una ontología basada en el conocimiento contenido en la guía de SWEBOK. El objetivo es dar fácil acceso al conocimiento de la guía de SWEBOK. Con ese conocimiento es posible ayudar a la definición y mejora de procesos dentro de las empresas.

2. Desarrollo del sistema software. Este sistema es derivado al trabajo de la ontología. Se trata de un sistema que permite la lectura y razonamiento de modelos ontológicos. El sistema está diseñado para leer cualquier ontología. Pero su funcionalidad está orientada al uso de la ontología de SWEBOK. Este sistema software permite la utilización del conocimiento contenido dentro de la ontología. Además tiene la capacidad de almacenar de manera persistente la información relacionada a las operaciones de la empresa. El almacenamiento se hace dentro de una base de datos relacional (RDB, del inglés Relational Data Base). Además, el sistema software permite a usuarios sin conocimientos avanzados en el uso de herramientas ontológicas poder crear y manejar tres aspectos claves para la gestión del conocimiento como son: ontologías, inteligencia y datos dinámicos.

En resumen, el sistema está basado en la ontología de SWEBOK (aunque puede leer cualquier modelo ontológico, las excepciones de manejo y lectura de ontologías son mostradas en la sección 4.2) y almacena los datos de la ontología en una RDB.

2.1. Usuarios Finales

La estructura de la guía de SWEBOK está dividida en 10 áreas de conocimiento; la ontología aquí presentada, mantiene esa modularidad. Esto permite que la ontología pueda adaptarse a distintos requerimientos. De igual manera el uso de la ontología se puede adaptar a través del sistema software. Todo esto mediante los aspectos que puedan ser requeridos en diferentes tamaños y complejidades de proyectos.

1. PyMEs hasta empresas de gran tamaño que realizan proyectos de software de una complejidad considerable. La ontología y el sistema brindan el conocimiento necesario para poder definir y/o mejorar su definición de proceso software, definición de roles, desarrollo de políticas de evaluación y desempeño, especificar tareas de desarrollo software, implementar las prácticas establecidas por el SWEBOK y brindar fácil acceso a la información contenida en el SWEBOK.
2. Grupos de trabajo de universidades que requieran de la utilización de SWEBOK, puede ayudar a definir requerimientos de educación, formación y desarrollo de políticas de evaluación de desempeño.
3. Usuarios sin conocimientos avanzados en SWEBOK ni en el uso de herramientas ontológicas. Las propuestas contenidas en el presente

trabajo ayudan a este tipo de usuarios al leer una ontología y a hacer uso de la información contenida dentro de esta (ontología), sin la necesidad de tener conocimientos avanzados sobre ontologías .

4. En general cualquier organización y usuarios que en la actualidad implementa o desee implementar las prácticas establecidas por el SWEBOK.
- 5.

3. Metodología

El desarrollo de la ontología está basado en una metodología sistemática. Esta metodología, a su vez, está basada en las metodologías Methontology (Corcho, Fernández, Gómez & López, 2005) y On-To-Knowledge (Sure, Staab & Studer, 2002), ambas estructuradas dentro del ciclo de mejora continua (Muñoz, Espuña & Puigjaner 2010) o ciclo PDCA. El ciclo PDCA (Plan, Do, Check and Act), es un modelo de gestión de calidad clásica (Ning, Cheng & Liu, 2010; Zhichun, 2011). La descripción de esta metodología puede verse más a detalle en la sección 4.2 y en la Figura 1. Por otro lado, dentro del desarrollo del prototipo del sistema software, la metodología se basó en el desarrollo ágil planteado por SCRUM .

4. Desarrollo

En la actualidad, existen numerosos estándares para la ingeniería de software y procesos de producción. Este tipo de estándares, generalmente, tratan de cubrir todos los aspectos que están involucrados dentro de cierta área de conocimiento o de aplicación práctica en la industria. La cantidad de información y conceptos que contienen estos documentos puede representar una dificultad en su manejo; además, se le puede añadir a esta dificultad, la cantidad de información que es propia de la organización. Con respecto a esto, como una solución, las ontologías ayudan a la mejora de procesos de desarrollo de software. Así, ayudan en el entendimiento común de la información entre miembros del equipo de desarrollo de software (Mendes & Abran, 2004; Willie, Abran, Desharnais & Dumke 2004; Bermejo, 2006), así

como en facilitar el análisis del estado del proceso que actualmente ejecutan, identificar oportunidades de mejora y plantear metas de mejora. En este contexto, existen ontologías que han ayudado en la mejora de procesos software como lo son la ontología de Capability Maturity Model Integration for Development (CMMI-DEV) (Soydan, 2012). Existen además ontologías híbridas: la ontología que integra Personal Software Process (PSP) y herramientas utilizadas por Six Sigma (Kim et al., 2009), la ontología que integra CMMI con el estándar ISO 9001:2000 (Ferchichi, Bigand & Lefebvre, 2008) y la que une Six Sigma con Information Technology Infrastructure Library (ITIL) (Dash, 2012).

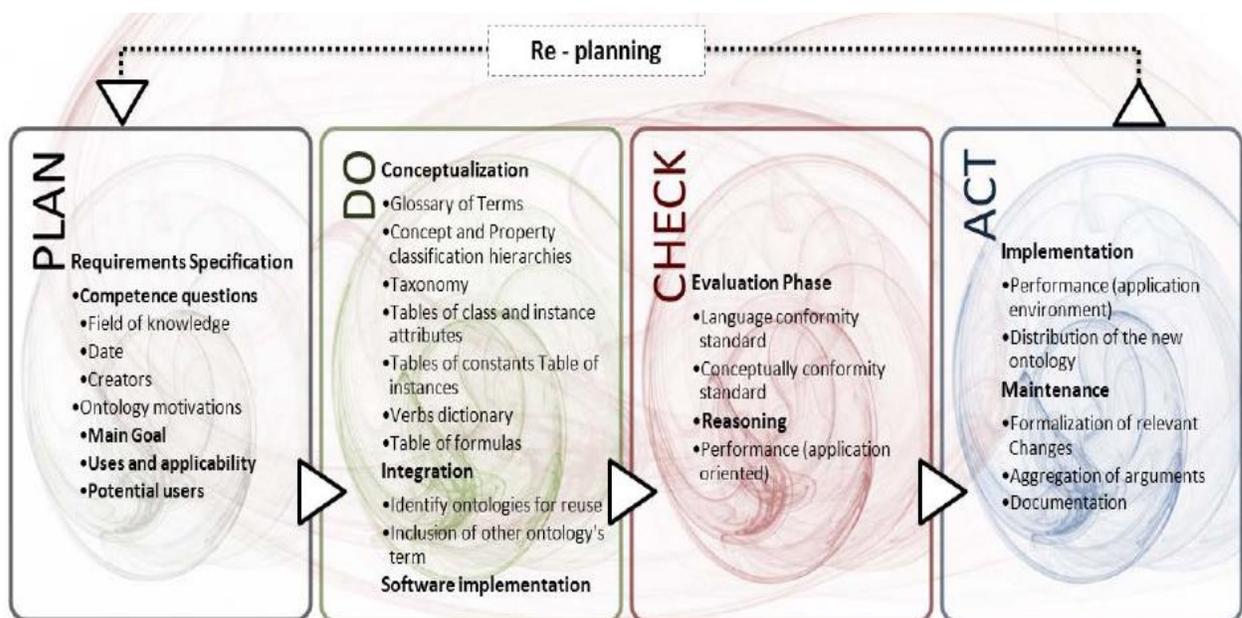


Figura 1. Metodología propuesta para el desarrollo de la ontología SWEBOK (Muñoz, Mejía, Muñoz, 2013).

4.1 Cuerpo de Conocimiento de la Ingeniería de Software (SWEBOK)

La guía de SWEBOK está compuesta de 10 áreas de conocimiento (Mendes & Abran, 2004), como se puede ver en la Figura 2 y tiene como objetivos: caracterizar los contenidos de la disciplina de la ingeniería de software (IS), proveer un acceso típico al cuerpo de conocimiento de la IS, proveer una vista consistente mundial de la IS, clarificar el lugar de la IS y fijar sus límites y proveer fundamentos para desarrollo de plan de estudios y de material de certificación individual (Willie, Abran, Desharnais & Dumke., 2003, 2004; Mendes & Abran, 2004; Abran et al., 2006).

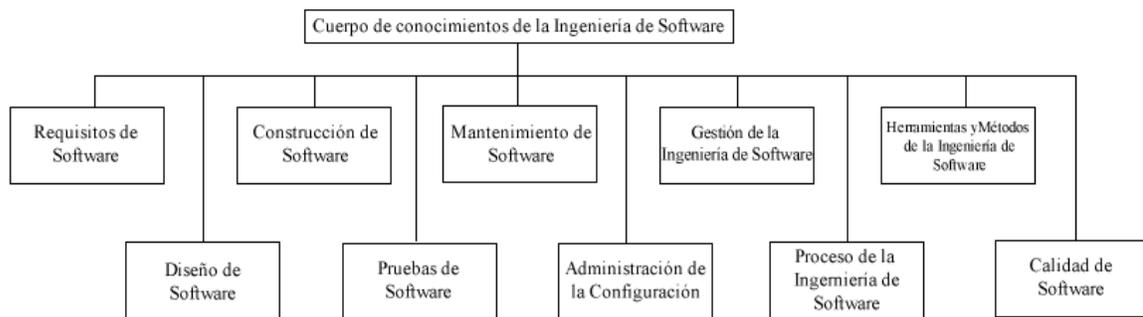


Figura 2. Áreas del Conocimientos contenidas dentro del Cuerpo de Conocimientos de la Ingeniería de Software.

4.2 Desarrollo de la Ontología de SWEBOK

Para la construcción de la ontología se siguen los pasos de la metodología propuesta:

Fase de Planeación. Investigación de la situación actual del dominio mediante la lectura de la guía de SWEBOK en (Abran, Bourque, Dupuis, Moore & Tripp, 2004). Derivado a esto, se hizo:

1. Una estrategia de desarrollo. Se plantean 6 actividades principales:
 1. Revisión literaria. Esta actividad significa la búsqueda de literatura relacionada con el dominio.
 2. Estudio del dominio. Se refiere al siguiente paso después de revisar la literatura con el fin de comprender plenamente el dominio.
 3. Revisión de la metodología. Para el desarrollo de la ontología, la metodología debe ser revisada. Esta revisión permitirá seguir los pasos adecuados para la correcta interpretación de la ontología.
 4. Implementación de la metodología. Después de revisar minuciosamente la metodología, el siguiente paso es seguir de una manera secuencial los pasos de esa metodología para el desarrollo de la ontología.
 5. Entregables del proyecto. Esto se refiere a los documentos y archivos entregables (de acuerdo a la calendarización del proyecto) relacionados con la ontología.
 6. Repetir las actividades desde la actividad 1 a la 6. Este es un proceso incremental e iterativo. Con el fin de lograr los resultados deseados es necesario el perfeccionamiento de las actividades.
2. Identificación del propósito. El propósito es construir una ontología para ayudar en el uso del SWEBOK y procesos de IS.
3. Usuarios finales. Los usuarios finales son mostrados en la sección 2.1.
4. Definición de dominio. La definición de dominio se encuentra en la sección 4.1.
5. Definición de alcance mediante 11 preguntas de competencia (las preguntas de competencia, son preguntas que la ontología ya construida debe de ser capaz de responder) y son las siguientes:
 1. ¿Qué es la Ingeniería de Software?
 2. ¿Cuáles son las áreas de conocimiento de SWEBOK?
 3. ¿Qué disciplinas están relacionadas con la Ingeniería de Software?
 4. ¿Qué herramientas y métodos se utilizan en Ingeniería de Software?
 5. ¿Cuáles son las características de la profesión de ingeniería del software?
 6. ¿Cuáles son los fundamentos en la construcción de software?
 7. ¿Cuáles son los fundamentos de diseño de software?
 8. ¿Cuáles son los fundamentos de requerimientos de software?
 9. ¿Cuáles son los fundamentos de pruebas de software?
 10. ¿Cuáles son los fundamentos del mantenimiento de software?
 11. ¿Cuáles son los fundamentos de calidad de software?

Fase de Desarrollo. Creación del glosario de términos. Se identifican 12 clases principales, 403 conceptos, tabla de verbos y 471 propiedades de objeto. Con el fin de la creación de la taxonomía se engloban cada una de las 403 clases dentro de una clase principal. La taxonomía es mostrada en la Figura 3.

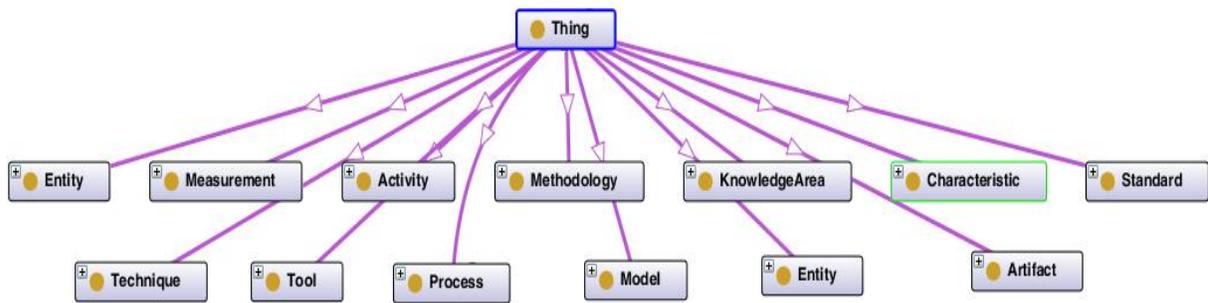


Figura 3. Taxonomía de la Ontología de SWEBOK.

Fase de revisión. La conceptualización de la ontología se ha verificado mediante el uso de axiomas experimentales y la introducción de instancias a través de la herramienta Protégé 4.1 . Se encontraron algunas inconsistencias, 11 subclases de la taxonomía se encontraban englobadas en la clase principal incorrecta.

Fase de Actuación. Se tomaron las acciones correctivas sugeridas en la fase anterior (fase de revisión).

Fase de Re planeación. Con el fin de ayudar en el ambiente empresarial, otras preguntas de competencia, axiomas y reglas necesitan ser planteadas para satisfacer las necesidades de mejora de definición de procesos de la compañía. En esta etapa, el uso del sistema software, 4 preguntas de competencia, 2 propiedades de objeto, 1 clase y 32 reglas son propuestas.

4.2 Desarrollo del sistema para el manejo de modelos ontológicos

El desarrollo del sistema software está enfocado a la lectura y razonamiento de una ontología modelada en lenguaje OWL (Web Ontology Language). La construcción del sistema software se basó en el desarrollo ágil guiado por la técnica SCRUM. Por otra parte Java fue empleado como lenguaje de programación, en donde se permitía la implementación y el uso de las librerías OWLAPI 3.4.3 , Pellet 2.3.0 y HermiT 2.0 , todas ellas desarrolladas en Java. Además, adicionalmente en este trabajo se han probado de manera experimental potenciales herramientas y librerías que pudiesen ser de gran ayuda para el manejo de información y razonamiento de los modelos ontológicos. Por otro lado los formatos que el sistema software soporta son RDF, OWL y Turtle. Por el contrario, los archivos en formato KRSS2 y Manchester OWL no son garantizados que se carguen correctamente dentro del sistema software propuesto (de igual manera la herramienta Protégé en sus versiones 4.1, 4.2 y 4.3 tampoco puede leer estos formatos).

El sistema software ha sido probado mediante un caso de estudio (mostrado en la sección 4.3). Posteriormente, al caso de estudio, se hizo la integración de la herramienta con el marco o framework OWLDB. OWLDB utiliza como principal medio de almacenamiento a Hybernate. Esta aplicación permite crear automáticamente una base de datos siguiendo la estructura ontológica. Durante la integración de OWLDB se tuvo que hacer un degradado con las librerías OWLAPI de 3.4.3 a 3.2.3 empleadas por el sistema, ya que OWLDB utiliza librerías OWLAPI 3.2.3, sacrificando así valiosas capacidades en inferencia y razonamiento.

4.3 Caso de Estudio

Con el fin de probar la funcionalidad de la ontología y el sistema software se brinda una solución práctica mediante un caso de estudio en una compañía.

4.3.1 Entorno de aplicación

La compañía consiste de 40 empleados compuestos por administradores, diseñadores y personal de apoyo. El estatus actual de la compañía fue definido por medio de entrevistas con el personal que trabaja en tres diferentes niveles de la compañía.

4.3.2 Hallazgos

Por medio de las entrevistas fueron encontrados varios problemas, estos incluyen: falta de definición de procesos de desarrollo software, falta de estándares y modelos de calidad. La etapa de pruebas de la compañía no está bien formalizada, donde cuatro tareas principales son identificadas:

1. Revisión inicial.
2. Planeación.
3. Revisión de ciclos.
4. Revisión de los resultados.

De las cuatro tareas, únicamente la tarea de planeación puede ser descompuesta en cinco acciones (las demás tareas no pueden ser descompuestas debido a la falta de definición del proceso):

1. Revisión de requerimientos.
2. Revisión con el cliente.
3. Análisis de riesgos.
4. Definición de casos de prueba.
5. Planeación de pruebas.

Además tiene los roles: equipo de pruebas, usuario final y equipo de desarrollo.

4.3.3 Desarrollo

El caso de estudio ha sido probado específicamente para el proceso de pruebas de la empresa. Para probar el proceso se empleó el área de conocimiento de pruebas de software de la Ontología de SWEBOK en conjunto con el sistema software. Con el fin de alcanzar la mejora de proceso de pruebas en la compañía, es necesario plantear preguntas de competencia específicas a este caso. Los conceptos contenidos en la definición del proceso de pruebas de la compañía (etapas, tareas, acciones, roles, herramientas y recursos mostrados en la sección 4.3.2) son vistos como instancias dentro de la ontología. Estas instancias fueron ingresadas de manera manual dentro de la ontología identificando la clase (o concepto) a la cual pertenece. Por ejemplo, la acción de revisión de requerimientos (dentro del proceso de la compañía) se refiere al concepto Requirements Reviews contenido en la guía de SWEBOK. Por ende, la acción de revisión de resultados es una instancia de la clase RequirementsReviews dentro de la ontología.

4.3.4 Resultados del Caso de Estudio

La introducción de la ontología de SWEBOK ayudó a la compañía a empezar una definición formal de este proceso y su mejora continua. La ontología SWEBOK brinda información acerca de en qué punto se encuentra la definición del proceso. La mejora de definición de proceso es logrado por medio de satisfacer toda el área de conocimiento de pruebas de software de SWEBOK. Esto ayuda en el análisis de brechas, al proveer aquellos conceptos cubiertos actualmente por el proceso, y aquellos cuales el proceso no tiene y necesita cubrir. Así, el sistema muestra el porcentaje cubierto por el proceso respecto al SWEBOK y las clases de SWEBOK que el proceso no cubre en su definición. El proceso de la compañía en el actual caso de estudio cubre el 12.5% del área de conocimiento de pruebas de SWEBOK, además se hace la verificación de si el proceso contiene entradas y salidas. Brindando así, nuevas oportunidades de mejora. Los resultados se muestran en la Figura 4.

5. Conclusiones

En el presente trabajo se brindó la construcción de la ontología de SWEBOK y el desarrollo del sistema software. El caso de estudio ha sido probado específicamente para el proceso de pruebas de la empresa. Para probar el proceso se empleó el área de conocimiento de pruebas de software de la Ontología de SWEBOK en conjunto con el sistema software (propuestos en el presente documento). La ontología de SWEBOK fue probada mediante un caso de estudio dentro de una compañía de consultoría en tecnologías de información, principalmente enfocada en el desarrollo de proyectos web, multimedia y aplicaciones móviles. La situación actual de la compañía fue recabada mediante entrevistas.

La introducción de la ontología de SWEBOK ayudó a la compañía a empezar una definición formal de este proceso y su mejora continua. Con el fin de alcanzar la mejora de proceso de pruebas en la compañía, fue necesario plantear preguntas de competencia específicas a este caso. Como se mencionó anteriormente, la ontología SWEBOK brinda información acerca de en qué punto se encuentra la definición del proceso. La mejora de definición de proceso es logrado por medio de satisfacer toda el área de conocimiento de pruebas de software de SWEBOK. El proceso de la compañía en el actual caso de estudio cubre el 12.5% del área de conocimiento de pruebas de SWEBOK, además se hace la verificación de si el proceso contiene entradas y salidas. Brindando así, nuevas oportunidades de mejora.

Como se adelantó en la sección anterior con la integración de la herramienta y OWLDB, se logró almacenar la ontología de SWEBOK de manera persistente dentro de una base de datos relacional MySQL. OWLDB permite el almacenamiento de todo el esquema del modelo ontológico. Desde el manejador de MySQL, phpMyAdmin, se constató que se ha guardado todo el esquema dentro de 65 tablas, algunas de las tablas son mostradas en la Figura 5. Sin embargo los resultados no fueron del todo satisfactorios. Se logró

realizar el degradado con las librerías, pero se sacrificaron funcionalidades dentro de la herramienta, por lo que significa un retroceso al trabajo. OWLDB emplea Hibernate para construir y poblar las tablas automáticamente a la base de datos. Sin embargo este es un proceso que toma bastante tiempo. En pruebas realizadas (por medio de NetBeans 7.1.2 en una PC con 3.48 GB de RAM, Intel Core i5-2310 a 2.90 Ghz y Sistema Operativo de 64 bits) el almacenamiento de la ontología SWEBOK tardó 6 min. 35 seg. y 6 seg. de carga desde la BD a el sistema.

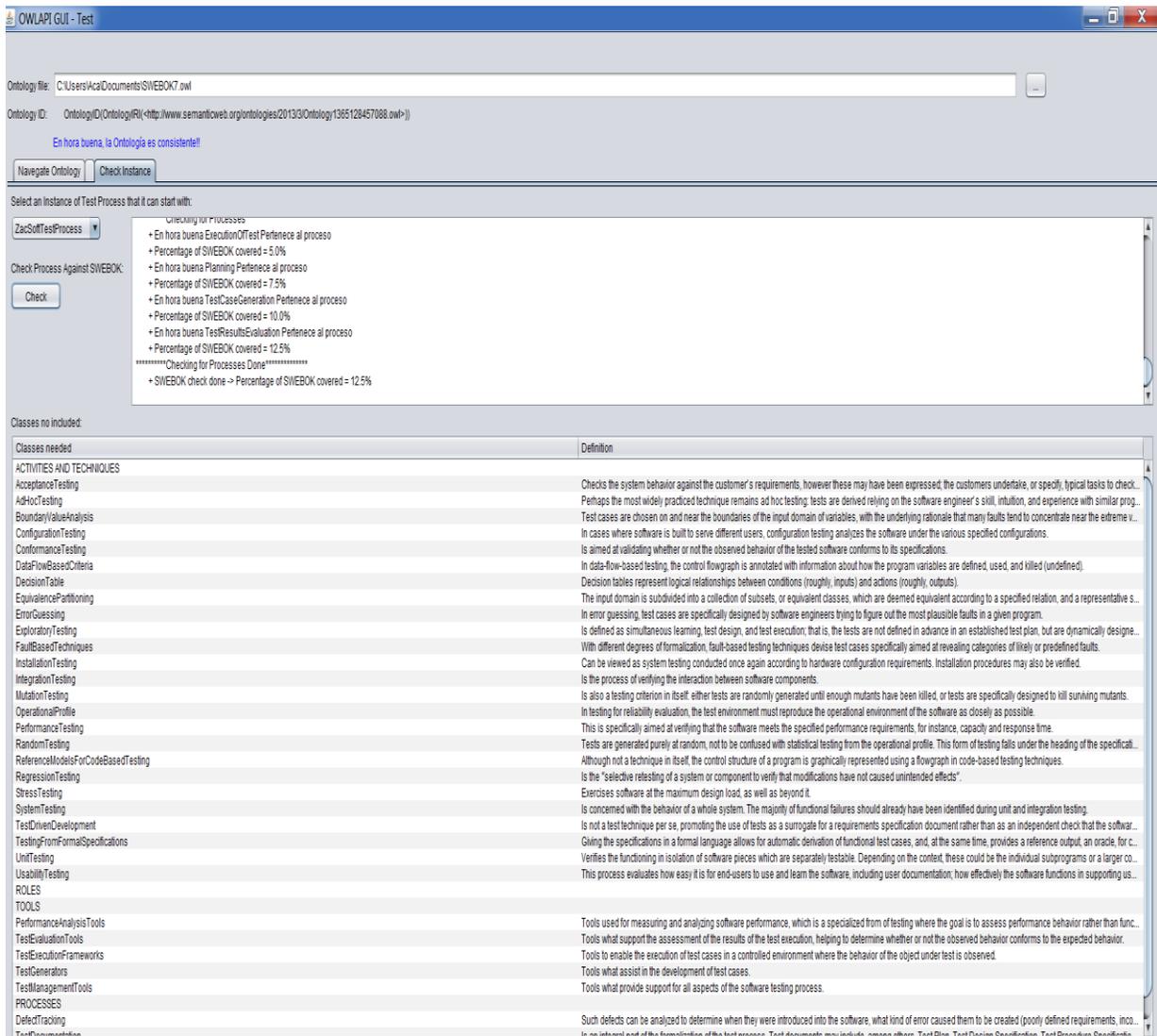


Figura 4. Sistema con la Ontología de SWEBOK mostrando los resultados del caso de estudio.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
annotationassertion	Examinar Estructura Buscar Insertar Vaciar Eliminar	409	InnoDB	latin1_swedish_ci	112 KB	-
annotationiri	Examinar Estructura Buscar Insertar Vaciar Eliminar	408	InnoDB	latin1_swedish_ci	96 KB	-
annotationproperty	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	latin1_swedish_ci	32 KB	-
annotationpropertydomain	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
annotationpropertyrange	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
annotation_annotations	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
classassertionaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	latin1_swedish_ci	64 KB	-
class_expressions	Examinar Estructura Buscar Insertar Vaciar Eliminar	454	InnoDB	latin1_swedish_ci	80 KB	-
class_operands	Examinar Estructura Buscar Insertar Vaciar Eliminar	441	InnoDB	latin1_swedish_ci	48 KB	-
dataoneof_values	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
datapropertycharacteristicaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
datarange_facets	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
declarationaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	941	InnoDB	latin1_swedish_ci	176 KB	-
disjointunionaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
facetrestriction	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
haskeyaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
importsdeclaration	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16 KB	-
individualrelationshipaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	42	InnoDB	latin1_swedish_ci	80 KB	-
indi_parts	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	48 KB	-
inverseobjectpropertiesaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	64 KB	-
key_prop_exprs	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
objectoneof_values	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
objectpropertychainsubpropertyaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
objectpropertycharacteristicaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	411	InnoDB	latin1_swedish_ci	48 KB	-
obpropchainsub_parts	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
ontologyiri	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	latin1_swedish_ci	16 KB	-
owlannotation	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	64 KB	-
owlanonymousindividual	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
owlcardinalityrestriction	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	latin1_swedish_ci	48 KB	-
owlclass	Examinar Estructura Buscar Insertar Vaciar Eliminar	418	InnoDB	latin1_swedish_ci	176 KB	-
owlconstant	Examinar Estructura Buscar Insertar Vaciar Eliminar	401	InnoDB	latin1_swedish_ci	128 KB	-
owldatacomplementof	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
owldataproperty	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	64 KB	-
owldatatype	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	latin1_swedish_ci	48 KB	-
owldatypedefinition	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	64 KB	-
owldatyperestriction	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
owlNamedindividual	Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	latin1_swedish_ci	48 KB	-

Figura 5. Base de Datos de la Ontología de SWEBOK.

6. Trabajo futuro

Actualmente se encuentra en proceso la investigación de un marco o framework llamado Ontop que permite mitigar los problemas presentados por OWLDB en almacenamiento en bases de datos. Al contrario de OWLDB, Ontop es un framework actual y permite consultas ágiles en la base de datos. Las pruebas con este framework han sido exitosas; de igual manera las pruebas con Jena y almacenamiento persistente también han sido exitosas. Por otra parte, también se encuentra en actual desarrollo la unión entre ontologías de SWEBOK, CMMI-DEV y Six Sigma. En un futuro se planea adaptar las librerías de Ontop y la unión de ontologías para adecuarlas a las necesidades de mejora de procesos de software por medio del sistema mostrado en el presente documento.

Referencias

Abran, A., Bourque, P., Dupuis, R., Moore, J., Tripp, L. (2004). Guide to the Software Engineering Body of Knowledge (SWEBOK). IEEE Press, Piscataway, NJ.

Abran, A., Cuadrado, J., García, E., Mendes, O., Sánchez, S. & Sicilia, M. (2006). Engineering the Ontology for Software Engineering Body of Knowledge: Issues and Techniques. En Ontologies for Software Engineering and Software Technology (pp. 103-121).

Aksoy, C., Alparslan, E., Bozdağ, S., Çulhacı, İ. (2011). OSDBQ: Ontology Supported RDBMS Querying. Metadata and Semantic Research. Communications in Computer and Information Science, Vol. 240, (pp. 47 – 55). Berlin, Alemania: Springer-Verlag. doi: 10.1007/978-3642-24731-6_5.

Barchini, G., Álvarez, M. (2010). Dimensiones e Indicadores de la Calidad de una Ontología. Revista Avances en Sistemas e Informática, 7 (1). Consultada el 3 de Octubre de 2012, de <http://www.redalyc.org/articulo.oa?id=133115523004>

Bermejo, A. (2006). *Ontology-based Software Engineering, Engineering Support for Autonomous Systems. Integrating Cognition+Emotion+Autonomy*, European Integrated Project IST-027819.

CMMI Product Team. (2010). *CMMI® for Development, Version 1.3 (CMU/SEI-2010-TR-033)*. Consultado de http://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf

Corcho, O., Fernández, M., Gómez, A. & López, A. (2005). Building legal ontologies with METHONTOLOGY and WebODE. *Law and the Semantic Web* (pp. 142-157). doi: 10.1007/978-3-540-322553-5_9.

Culot, N., Ghawi, R., Yétongnon, K. (2007). DB2OWL : A Tool for Automatic Database-to-Ontology Mapping [Versión electrónica]. SEBD. P. 491-494. Consultada el 6 de Octubre de 2012, de <http://www.citeulike.org/user/stephane-jean/article/2424618>

Dash, S. (2012). Ontology Driven Benchmarking in ITIL to achieve Six Sigma. *International Journal of Computer Applications*, (0975-8887), 42 (19). Consultada el 16 de Octubre de 2012, de <http://research.ijcaonline.org/volume42/number19/pxc3877826.pdf>

Ferchici, A., Bigand, M., Lefebvre, H. (2008). An Ontology for Quality Standards Integration in Software Collaborative Projects. En *Proceedings of the First International Workshop on Model Driven Interoperability for Sustainable Information Systems. MDISIS'08*, p. 17-30. Montpellier, France. Consultada el 16 de Octubre de 2012 de <http://ceur-ws.org/Vol-340/paper02.pdf>

Heymans S., Ma, L., Anicic, D., Ma, Z., Steinmetz, N., Pan, Y., Mei, J., Fokoue, A., Kalyanpur, A., Kershenbaum, A., Schonberg, E., Srinivas, K., Feier, C., Hench, G., Wetzstein, B., Keller, U. (2008). Ontology Reasoning with Large Data Repositories. *Ontology Management. Computing for Human Experience*, Vol. 7, (pp. 89-128). Berlin, Alemania: Springer-Verlag. doi: 10.1007/978-0-387-69900-4_4.

Kim, N., Andrews P., Asselbergs, F., Frost, H., Williams, S., Harris, B., Read, C., Askland, K. & Moore, J. (2012). Gene Ontology analysis of pairwise genetic associations in two genome-wide studies of sporadic ALS. *BioData Mining* 9(5).

Mendes, O. & Abran, A. (2004). Software Engineering Ontology: A Development Methodology. *Metrics News* 9(1) p. 64-71.

Muñoz, E., España, A. & Puigjanier, L. (2010). Towards an Ontological Infrastructure for Chemical Batch Process Management. *Computers & Chemical Engineering*, 34 (5), p. 668-682.

Muñoz, M, Mejía, J. & Muñoz, E. (2013). Knowledge Management to Support using Multi-model Environments in Software Process Improvement. En 20th EuroSPI Conference. Dundalk Institute of Technology, Ireland.

Ning, J., Chen, Z. & Liu, G. (2010). PDCA Process Application in the Continuous Improvement of Software Quality. College of Computer Science and Engineering, Changchun University of Technology, p. 61–65, 201.

Pyzdek, T. (2003). The Six Sigma Handbook. A Complete Guide for Green Belts, Black Belts, and Managers at All Levels. [Adobe Digital Editions version]. doi: 10.1036/0071415963.

Quintanilla, A. (2005). Tecnología: In enfoque Filosófico y Otros Ensayos de Filosofía de la Tecnología (1ª Edición). México, D.F. Fondo de Cultura Económica, p. 65.

Romá, M. (2009). Tesis Doctoral. OntoFIS: Tecnología Ontología en el dominio farmacoterapéutico [Versión Electrónica]. Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. Consultada el 6 de Octubre de 2012, de http://gpsi.dlsi.ua.es/gpsii1/sites/default/files/tesis_Roma_OntoFIS.pdf

Soydan, G. & Kokar, M. (2012). A Partial Formalization of the CMMI-DEV – a Capability Maturity Model for Development. Journal of Software Engineering and Applications, 5 (10), p. 1-25.

Sure, Y., Staab, S. & Studer, R. (2002). Methodology for development and employment of ontology based knowledge management application. ACM Sigmod Record 31 (4) p. 18-23. doi: 10.1145/637411.637414.

Trinkunas, J. & Vasilecas, O. (2007). Building Ontologies from Relational Databases using Reverse Engineering Methods. En Rachev B., Smrikarov, A. & Dimov, D. (Eds.), Proceedings CompSysTech '07 Proceedings of the 2007 international conference on Computer systems and technologies. Bulgaria. doi: 10.1145/1330598.1330614.

Wille, C., Abran, A., Desharnais, J. & Dumke, R. (2003). The Quality concepts and sub-concepts in SWEBOK: An Ontology challenge. En International Workshop on Software Measurement (IWSM), Montreal, Canadá, p. 18.

Wille, C., Abran, A., Desharnais, J. & Dumke, R. (2004). E-Learning Infrastructure for Software Engineering Education: Steps in Ontology Modeling for SWEBOK. En Proceedings of the IASTED International Conference SOFTWARE ENGINEERING, Innsbruck, Austria.

Zhichun, Q. (2011). Quality improvement of wall energy conservation project based on PDCA cycle, 2011 International Conference on Electric Technology and Civil Engineering (ICETCE), p. 1416–1419.

Notas biográficas:



José Eduardo Guadalupe Gaytán Solís Solís es actual estudiante de la maestría en ingeniería de software en el Centro de Investigación en Matemáticas A.C. (CIMAT), unidad Zacatecas, en Zacatecas, México. Obtuvo el grado de Ingeniero en Computación en el año 2012 en la Universidad Autónoma de Zacatecas. Cuenta con las certificaciones: Microsoft MCTS .NET Framework 2.0, Windows Development Foundation, Microsoft MCTS .NET Framework 3.5, ASP .NET Applications, Oracle Java Standard Edition 6 Programmer, Novel Certified Linux Desktop Administrator SUSE Linux Enterprise Desktop 10 y SEI-Certified PSP Developer. Sus intereses son programación de aplicaciones web y de escritorio, inteligencia artificial y arquitectura de software. En deportes: Fútbol Soccer, Fútbol Americano, Atletismo.



Ricardo González Saldívar Saldívar es actual estudiante de la Maestría en Ingeniería de Software en el Centro de Investigación en Matemáticas A.C. (CIMAT), unidad Zacatecas, en Zacatecas, México. Obtuvo el grado de Ingeniero en Sistemas Computacionales en el año 2010 en el Instituto Tecnológico Superior de Nochistlán. Cuenta con las certificaciones: Sun Certified Java SE Developer y SEI-Certified PSP Developer. Sus intereses son Arquitectura y Diseño de Software, Procesos de Desarrollo de Software, Ciencias de la Computación y Gestión del Conocimiento. Le gusta practicar ajedrez, ilustración digital y la lectura.



Edrisi Muñoz Mata Ingeniero industrial con especialidad en manufactura y Maestro en ciencias en ingeniería industrial con especialidad en calidad por el Instituto tecnológico de Orizaba (ITO) de México. Doctor en filosofía en ingeniería de procesos químicos por la Universidad Politécnica (UPC) de Cataluña de España. Su área de investigación principal es la gestión del conocimiento mediante el desarrollo de modelos ontológicos, sistemas de soporte a las decisiones en distintas áreas de proceso y optimización de procesos mediante el uso de modelos analíticos rigurosos. Actualmente es investigador asociado del Centro de Investigación en Matemáticas A.C. (CIMAT) de México, así como investigador invitado en el Centro de procesos y medio ambiente en la UPC. Su participación compete diferentes proyectos de investigación mexicanos y europeos. Participa en la publicación de diferentes artículos en revistas internacionales indexadas, así como en distintos congresos internacionales de renombre. Última publicación: Edrisi Muñoz, Elisabet Capón-García, José Miguel Laínez, Antonio Espuña, Luis Puigjaner, Integration of enterprise levels based on an ontological framework, Chemical Engineering Research and Design, Volume 91, Issue 8, August 2013, Pages 1542-1556, ISSN 0263-8762.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Selección de estrategias para la implementación de Mejoras de Procesos Software

Brenda Liliana Durón del Villar

**Centro de Investigación en Matemáticas
Av. Universidad No. 222, 98068 Zacatecas, México
brenda.duron@cimat.mx**

Mirna Ariadna Muñoz Mata

**Centro de Investigación en Matemáticas
Av. Universidad No. 222, 98068 Zacatecas, México
mirna.munoz@cimat.mx**

Resumen: La mejora de procesos software se ha convertido en la forma más lógica y obvia de dirigir la creciente necesidad de aumentar la competitividad en las empresas de desarrollo de software. Desafortunadamente no todas las implementaciones de mejoras en los procesos tienen el rendimiento deseado, debido a que los modelos y estándares existentes centran su atención en qué actividades implementar sin abordar el cómo implementarlas. Sin embargo, la identificación de qué actividades implementar no es suficiente y el conocimiento del cómo implementarlas es requerido para el éxito de la implementación de iniciativas de Mejoras de Procesos Software (MPS) en las empresas de desarrollo de software. En este trabajo se presenta los resultados tanto de una revisión del estado actual en la implementación de MPS lograda mediante una revisión literaria así como los resultados de un análisis de empresas locales de Zacatecas logrado mediante la aplicación de entrevistas. Finalmente, se presenta una propuesta para el establecimiento de estrategias para la

implementación de MPS basadas en los aspectos contextuales en los cual el software es desarrollado y entregado, tal que, pueda ser seleccionada una estrategia acorde a las necesidades y cultura de trabajo de la organización.

Palabras clave: Mejoras de procesos software, Estrategias de Implementación, empresas desarrollo de software.

Selection of strategies for the implementation of Software Process Improvement

Abstract: Software process improvement has become the most logical and obvious way to address the growing need of increasing the competitiveness in software development companies. Unfortunately not all process improvements implementations have the desired results, because existing models and standards focus their attention on what activities implemented without addressing how to implement them. However, identifying what activities to implement is not enough; the knowledge of how to implement them is required for a successful implementation of Software Process Improvement (SPI) initiatives in organizations. This paper presents the results of reviewing both the current state of the implementation of SPI using a literature review and by performing interviews in software development organizations of Zacatecas. Finally, based on the findings of this analysis, the paper includes a proposal to develop strategies for the implementation of SPI based on the contextual aspects in which the software is developed and delivered, so that the strategy according to the organization needs and work culture can be selected.

Keywords: Software Process Improvement, Implementation strategies, software development organizations.

1. Introducción

Para una empresa de desarrollo de software es de vital importancia crear estrategias para ser competitiva frente a sus competidores. Es por esto que, las organizaciones implementan iniciativas de Mejoras de Procesos Software para incrementar la calidad del producto reduciendo el tiempo de entrega y los costos de producción (Cugola & Ghezzi, 1998) con lo cual consiguen esa ventaja competitiva. Como consecuencia de esto, la mejora de procesos de software se ha convertido en un mecanismo importante ya que impulsa la competitividad y eficiencia de las organizaciones de software (Cuevas et al., 2002).

En este contexto, en los últimos años se han desarrollado modelos y estándares para la apoyar a las organizaciones en la implementación de mejoras de los procesos de desarrollo de software, como el modelo CMMI, o la norma ISO/IEC15504, estos modelos y estándares centran su atención en qué actividades implementar sin abordar el cómo implementar estas actividades. No obstante, el conocimiento de cómo implementarlos es requerido para una exitosa implementación de estas iniciativas de Mejora de Procesos Software (MPS) en las empresas. Por lo que, un problema actual con la mejora de procesos de software no es la falta de estándares o modelos, si no la falta de estrategias efectivas de implementación de esos estándares o modelos (Niazi, Wilson & Zowghi, 2005).

Es en este contexto y con el objetivo de encontrar el cómo realizar una implementación exitosa de MPS se realiza esta investigación donde se pretende establecer y proponer un conjunto de estrategias tal que las organizaciones puedan elegir la que mejor les sirva teniendo en cuenta su contexto organizacional. Este trabajo se organiza de la siguiente manera, la sección 2 presenta el estado actual de MPS; la sección 3 muestra una investigación de MPS realizada en el ámbito local; la sección 4 presenta la propuesta de estrategias para una implementación de MPS y finalmente en la sección 5 se muestran algunas conclusiones y trabajo futuro.

2. Estado Actual de Mejora de Procesos Software

Para conocer el estado actual de MPS se realizó una Revisión Sistemática (RS) en 5 de las principales fuentes electrónicas de artículos científicos. Una revisión sistemática de la literatura permite identificar, evaluar, interpretar y sintetizar todas las investigaciones existentes y relevantes en un tema de interés particular (Feiler & Humphrey, 1992); por esto los resultados de la revisión sistemática se tomaron como punto de partida para esta investigación. En la Figura 1, se muestra el procedimiento para selección de estudios.

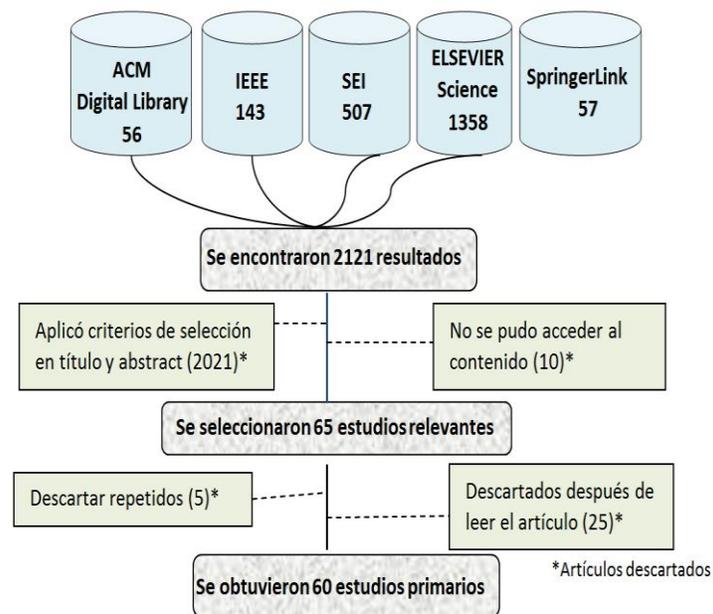


Figura 1. Selección de estudios de la Revisión Sistemática

2.1 Análisis de los Resultados

Estudios realizados muestran que el 67% de los gestores solicitan orientación sobre cómo implementar las iniciativas de mejoras de procesos software, en lugar de qué actividades implementar (Herbsleb & Goldenson, 1996). Lo que ha causado el aumento de la investigación en materia de implementación de mejoras de procesos en los últimos años.

Dentro del análisis de resultados se identificaron los siguientes problemas comunes en las estrategias de implementación de MPS: no se tienen claros los objetivos de la MPS; no se presta suficiente atención a los factores organizacionales que promueven o inhiben las mejoras de procesos y no se toma en cuenta el capital humano, y por lo tanto, está propenso a un inesperado o indeseado rendimiento y/o comportamiento. Se encontraron además que varios autores destacan listas de barreras y desmotivadores en las actividades de MPS (Niazi, Babar & Verner, 2010; Baddoo & Hall, 2003), siendo las más recurrentes las que se muestran en la Tabla 1.

Existen también varios estudios de campo así como investigaciones donde se busca encontrar los factores críticos para una exitosa implementación de MPS esto con la finalidad de garantizar que las iniciativas de implementación sean exitosas (Dyba Tore 2003; Montoni & Rocha 2010). Dentro de los factores identificados se encuentran la orientación de negocios, líderes y gerencias comprometidos e involucrados en el proceso, participación activa de los empleados, uso de medidas y capacitación y conciencia de la organización. Finalmente se identificaron y analizaron los métodos/modelos/frameworks existentes y se analizaron sus características.

Barreras	Desmotivadores
❖ Entendimiento del proceso de mejoras	❖ Falta de recursos dedicados a la implementación de mejoras
❖ Estructura organizacional y política	❖ Restricciones de tiempo y calendario
❖ Falta de gestión del proyecto de mejoras	❖ Experiencias negativas previas
❖ Falta de una metodología definida para la implementación de mejoras	❖ Falta de evidencia de los beneficios
❖ Falta de comunicación	❖ Iniciativas de mejoras impuestas
❖ Falta de soporte, herramientas y formación	

Tabla 1. Barreras y desmotivadores en la implementación de Mejora de Procesos Software encontrados en la Revisión Sistemática.

3. Investigación en Empresas Locales

3.1 Resultados

Como parte de este trabajo de investigación se realizaron entrevistas con empresas locales desarrolladoras de software de Zacatecas con la finalidad de contrastar con resultados reales los resultados obtenidos en la revisión sistemática. Para obtener la información de las empresas locales se elaboró un cuestionario como guía en las entrevistas para poder obtener la las experiencias en MPS de las empresas.

A continuación se listan los resultados de la investigación de las empresas locales enfocando en características, hallazgos, necesidades, mejores prácticas:

a) Características

Como resultado se han entrevistado 8 empresas de Zacatecas. En la Figura 2 se muestran las características de ellas basado en servicios que ofrecen, empleados por empresa y experiencia en MPS.

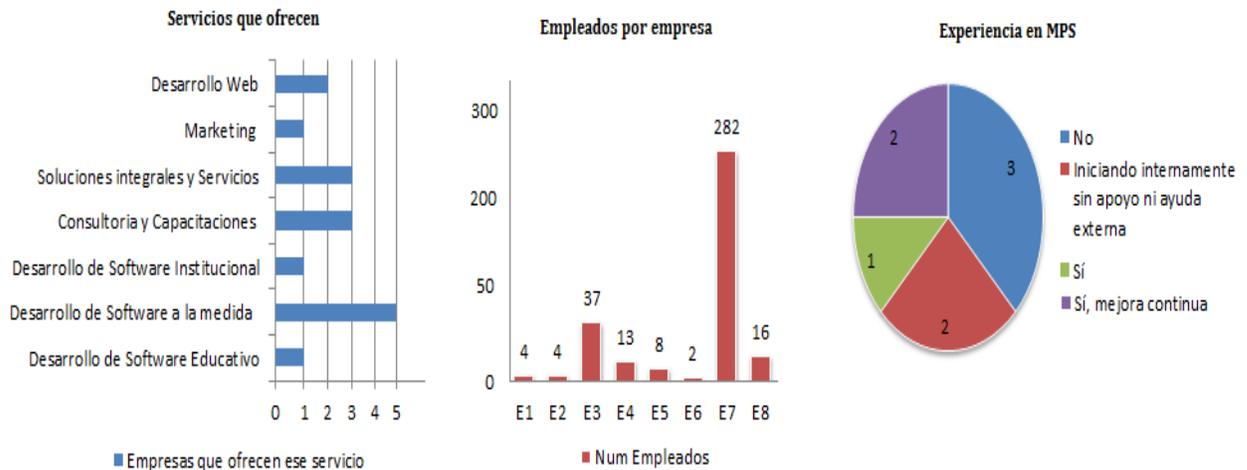


Figura 2. Caracterización de empresas encuestadas

Como se observa en la Figura 2: Como parte de los resultados además se identificaron hallazgos, necesidades y mejores prácticas de las empresas como se lista a continuación.

b) Hallazgos

- Solamente 3 de las empresas desarrolladoras de software analizadas cuentan con experiencia en MPS.
- Dos empresas están en un ciclo de mejora continua.
- Las dos empresas restantes apenas están reconociendo esta necesidad de mejorar.
- La empresa restante no considera necesario en su ámbito la adopción de algún modelo o método ya que consideran que no se adecuan a su forma de trabajo.

- Por otra parte ,cabe resaltar que existe un conocimiento pobre de un enfoque de procesos, por lo que no se cree que sea necesario implementar modelos y estándares aunado a que las empresas no cuentan con presupuesto suficiente para actividades de mejora lo que impide que estas empresas puedan obtener los beneficios de la mejora de procesos.

c) Necesidades

Cabe resaltar, por un lado, que las empresas entrevistadas con poca experiencia o nula en MPS piden dos cosas principales: 1) presupuesto para poder asignar personal a la mejora de procesos y 2) tiempo para implementar proyectos de mejora.

d) Mejores prácticas

Por otro lado, como resultado de las entrevistas a empresas que ya han tenido experiencia en MPS se obtuvieron algunas prácticas que les han ayudado al éxito en su implementación de las que destacan:

- Establecer inicialmente los objetivos, definiendo por qué se quiere una implantación de mejora procesos.
- Convencer a la alta gerencia para que sean promotores, patrocinadores y guías de la implementación de mejoras.
- Visualizar el beneficio que se obtendrá con la mejora de procesos.
- Apostar por el capital intelectual de la empresa.

3.2 Contraste de resultados revisión literaria e investigación en empresas locales

En este apartado se muestran las Barreras y desmotivadores en la implementación de Mejora de Procesos Software encontrados en la Revisión

Sistemática y que coincidieron en las entrevistas del ámbito local, estas barreras y desmotivadores se encuentran en la Tabla 2.

Barreras	Desmotivadores
❖ Entendimiento del proceso de mejoras	❖ Falta de recursos dedicados a la implementación de mejoras
❖ Falta de una metodología definida para la implementación de mejoras	❖ Restricciones de tiempo y calendario
❖ Falta de soporte, herramientas y formación	❖ Falta de evidencia de los beneficios

Tabla 2. Coincidencia entre Barreras y desmotivadores para la implementación de MPS.

4. Propuesta de Estrategias para la Implementación de MPS

Derivado de las experiencias obtenidas en la revisión de literatura como en el estudio de pymes del ámbito local de Zacatecas, se identificaron diferentes estrategias que han guiado y servido en la implementación de MPS y que pueden ayudar en la parte del cómo realizar las actividades de mejora. Sin embargo, aun cuando se han reportado los beneficios de éstas, esto no garantiza que sean adecuadas para todas las empresas, es decir, no existe una regla única que nos asegure el éxito (Jeners, Clarke, O'Connor, Buglione & Lepmets, 2013).

De acuerdo con (Petersen & Wohlin, 2009), el proceso de desarrollo de software idealmente debería estar en armonía con el contexto en el cual el software es desarrollado y entregado. Esto significa que con el fin de establecer si una solución específica puede tener éxito, es necesario describir

el contexto tan completa y precisamente como sea posible para el caso de estudio. Este trabajo se enfoca en la elección de las estrategias correctas para implementar mejoras de procesos adecuadas a cada contexto específico como a continuación se describe:

Se inicia con prefase en la que se confirma, mediante un checklist, que la organización tienen identificado como trabajan, cuáles son los objetivos de negocio que busca alcanzar; cuál es su rendimiento y su objetivo al implementar la mejora de procesos. Posteriormente, la empresa realiza una comparación de su entorno con los entornos establecidos en la propuesta por medio de categorías y así se les pueda proporcionar aquella estrategia que se adecue a sus necesidades. Para el establecimiento del conjunto de estrategias se toman como base los seis aspectos contextuales propuestos (Petersen & Wohlin 2009), considerados como aspectos a tener en cuenta para una implementación de mejora de procesos en cualquier organización: producto, proceso, prácticas y técnicas, recurso humano, organización y mercado, estos aspectos se describen brevemente en la Tabla 3.

Producto	Proceso	Prácticas, Herramientas, Técnicas	Recurso humano	Organización	Mercado
Madurez	Actividades	Herramientas	Roles	Modelo de la	Número de
Calidad	Flujo de	Prácticas y	Experiencia	organización	clientes
Tamaño	trabajo	técnicas		Unidades de	Segmento del
Tipo de sistema	Artefactos			negocio	mercado
Personalización				Certificaciones	Estrategias
Lenguaje de programación				Distribución	Restricciones

Tabla 3. Aspectos contextuales en los cual el software es desarrollado y entregado.

Además, para lograr el establecimiento de este conjunto de estrategias, se analizaran los principales métodos/modelos de implementación de estrategias; las lecciones aprendidas y las barreras y desmotivadores identificados tanto en

la revisión literaria como en el estudio de las pymes de la región. El procedimiento propuesto para la selección de la estrategia se muestra en la Figura 3.

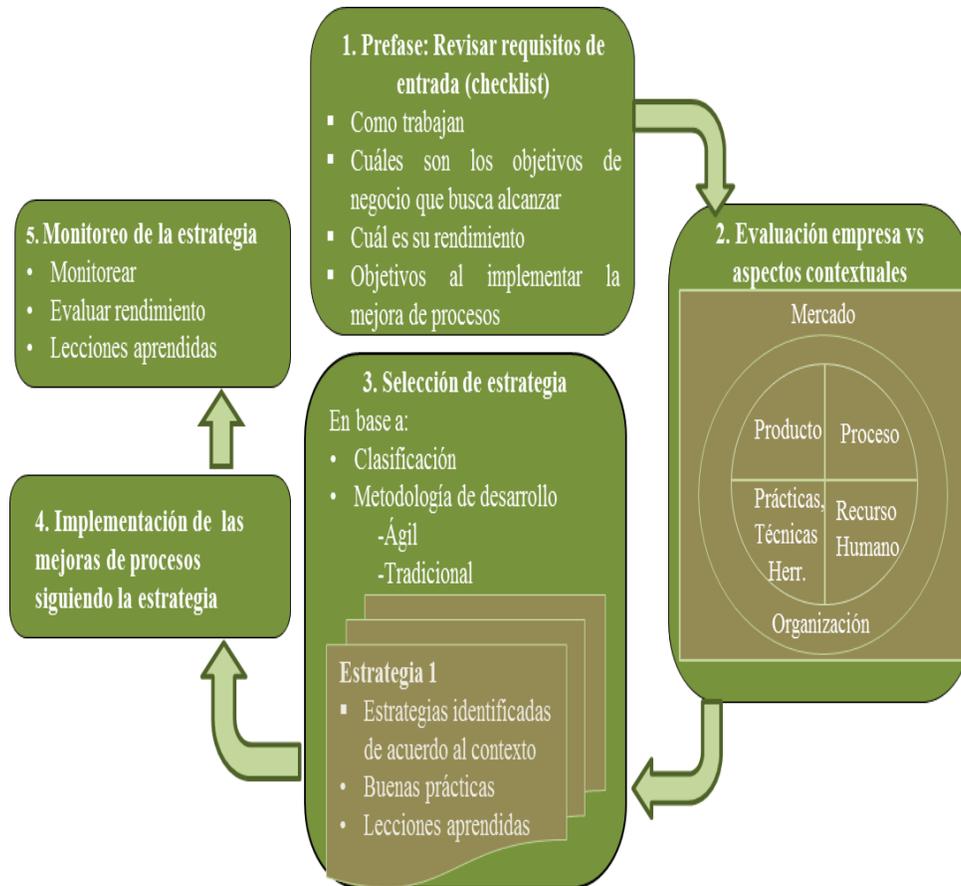


Figura 3. Propuesta de selección de estrategias para MPS.

5. Conclusiones y Trabajo Futuro

Las empresas han asimilado que la calidad del producto de software está directamente relacionada con la calidad del proceso utilizado para su desarrollo; y es por esto que, el interés en este campo de la ingeniería de software está en constante aumento reflejado en el desarrollo de nuevas técnicas, herramientas, metodologías y métodos que apoyan a las

organizaciones en el desarrollo de productos de calidad e implementación de mejoras en los procesos. Sin embargo, aún con la existencia de modelos, métodos y frameworks ya desarrollados para la implementación de mejoras y una serie de documentos en los que se identifican barreras, factores críticos para el éxito, así como motivadores y desmotivadores en la mejora de procesos software; sigue presentándose una falta de conocimiento en estrategias específicas enfocadas en cómo vencer todas esas barreras, como tratar los factores críticos.

Por lo tanto, se ha detectado además un nicho muy importante en la identificación de aquellas mejores prácticas de los modelos, métodos y estándares existentes que puedan marcar el camino que apoya en la conducción de una implementación de mejoras exitosa. Es importante resaltar que el éxito de MPS requiere de tolerancia al fracaso y que este fracaso es un prerrequisito esencial para el aprendizaje. Por lo tanto, las empresas deben ser capaces de convertir los fallos y problemas inesperados en oportunidades de aprendizaje.

Como trabajo futuro se vinculará cada aspecto contextual con categorías que permitan una comparación entre las empresas de una forma más sencilla. También se realizará el análisis de los modelos de implementación de MPS existentes identificados en la literatura y junto con las estrategias y lecciones aprendidas para crear el conjunto de estrategias de la propuesta. Una vez terminada la propuesta se probará su validez en empresas de desarrollo de software.

Referencias

- Baddoo Nathan, Hall Tracy (2003). De-motivators for software process improvement: an analysis of practitioners' views, *Journal of Systems and Software*, Volume 66, Issue 1, Pages 23-33
- Calvo-Manzano Jose A., Cuevas Gonzalo, Gómez Gerzón , Mejia Jezreel, Muñoz Mirna , San Feliu Tomás , (2009) Methodology for process improvement through basic components and focusing on the resistance to change, *Journal of Software Maintenance and evolution: Research and practice*.
- Cuevas G., De Amescua A., San Feliu T., Arcilla M., Cerrada J.A., Calvo-Manzano J.A., García M., (2002), *Gestión del Proceso Software*, Universitaria Ramon Areces, pp. 472.
- Cugola G. and Ghezzi C., (1998), *Software Processes: A Retrospective and a Path to the Future*, *Software Process: Improvement and Practice*, vol. 4, no. 3, pp. 101-12
- Dyba Tore (2003). Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context, *Proceedings of the 9th European software engineering conference*, Pages 148 - 157
- Feiler, P., Humphrey, W., (1992) *Software process development and enactment: Concepts and definitions*. SEI, Carnegie Mellon University, CMU/SEI-92-TR-004. Pittsburgh, Pennsylvania, USA
- Herbsleb, J. D. and Goldenson, D. R. (1996). A systematic survey of CMM experience and results. 18th international conference on software engineering (ICSE-18). Germany.
- Jeners, S., Clarke, P., O'Connor, R. V., Buglione, L., and Lepmets, M. *Harmonizing Software Development Processes with Software Development Settings – A Systematic Approach*, 20th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2013), CCIS Vol. 364
- Kai Petersen, Claes Wohlin ,(2009) *Context in Industrial Software Engineering Research*, Third International Symposium on Empirical Software Engineering and Measurement
- Montoni M.A., Rocha A.R. (2010). Applying Grounded Theory to Understand Software Process Improvement Implementation , *Seventh International Conference on the Quality of Information and Communications Technology*, Pages 25 - 34

Niazi Mahmood ,Wilson David ,Zowghi Didar (2005). A maturity model for the implementation of software process improvement: an empirical study, *Journal of Systems and Software*, 74, Issue 2, Pages 155–172.

Niazi Mahmood, Babar Muhammad Ali, Verner June M. (2010). Software Process Improvement barriers: A cross-cultural comparison, *Journal of Information and Software Technology*, Volume 52, Issue 11, Pages 1204-1216

Notas biográficas:



Brenda Liliana Durón del Villar recibió el grado de ingeniero en Computación de la Universidad Autónoma de Zacatecas; actualmente está estudiando la maestría en Ingeniería de Software en el Centro de Investigación en Matemáticas CIMAT.

Su interés es en la implementación exitosa de mejoras de procesos software así como las estrategias que ayuden a minimizar la resistencia al cambio en las empresas de desarrollo de software.



Mirna Ariadna Muñoz Mata, Doctor en Informática por la Universidad Politécnica de Madrid, en Madrid España, con mención de “Doctorado Europeo”. Ha realizado una estancia posdoctoral en la Universidad Carlos III de Madrid, España.

Actualmente es investigador del Centro de Investigación en Matemáticas (CIMAT) - Unidad Zacatecas en el área de Ingeniería de Software y es miembro del grupo de investigación Cátedra de Mejora de Procesos Software en el Espacio Iberoamericano (MPSEI), donde participa en proyectos internacionales de investigación con entidades educativas y de gobierno y de vinculación con la industria.

Ha participado en proyectos con la empresa everis consulting. Ha participado en el equipo de traducción oficial al español reconocida por el SEI del libro CMMI-DEV v1.2 y 1.3. Es miembro del comité científico de diversos congresos internacionales como: CISTI (2009-2012), del coloquio de investigación multidisciplinaria del Instituto Tecnológico de Orizaba (2011), del Infonor Chile 2012, del CERMA (2012) y de la revista internacional RISTI (2010-2012). Ha publicado diversos artículos técnicos en temas relacionados con la gestión de proyectos, implementación de mejora de procesos software, entornos multi-modelo y modelos y estándares de calidad. Es autora del libro Metodología Multimodelo para Implementar Mejoras de Procesos Software. Sus áreas de interés son: implementación de mejora de procesos software enfocando en la reducción de la resistencia al cambio, entornos multi-modelo, gestión del conocimiento y gestión del cambio en la mejora de procesos software, gestión de proyectos software, trabajo en equipo y modelos y estándares de calidad.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.