



GESTIÓN DEL CONOCIMIENTO ACERCA DE LOS PROCESOS DE DESARROLLO DE SOFTWARE EN LAS ASIGNATURAS DEL PROGRAMA DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD AUTÓNOMA DE MANIZALES

KNOWLEDGE MANAGEMENT IN SOFTWARE DEVELOPMENT PROCESSES IN SYSTEMS ENGINEERING COURSES AT THE MANIZALES AUTONOMA UNIVERSITY

Beatriz Ayala Hoyos, Mauricio Alba Castro y Herminia Quiceno Valencia

Universidad Autónoma de Manizales, Manizales (Colombia)

Resumen

Este artículo tiene como objetivo evidenciar la problemática vivida en la Universidad Autónoma de Manizales (UAM), específicamente en el Programa de Ingeniería de Sistemas, al no contar con un proceso de gestión del conocimiento para las labores de desarrollo de *software* que se llevan a cabo en las asignaturas Práctica Empresarial, Paz y Competitividad, Proceso de Investigación I y II, Diseño de Algoritmos, Programación Orientada a Objetos (POO), Énfasis Profesional, Gerencia de Proyectos Tecnológicos, Ingeniería de Software I y II, Proyecto de Desarrollo de Software y Bases de Datos, que son las que involucran en mayor proporción el desarrollo de *software*. También se percibe como problemática dentro del programa académico el desconocimiento de los elementos de gestión del conocimiento que deben permanecer al retirar o vincular un docente, con el fin de asegurar su saber específico requerido o generar un proceso de capacitación en lo relacionado con los procesos de desarrollo de *software*.

La investigación se inició con un diagnóstico de la situación actual de la gestión del conocimiento de los procesos de desarrollo de *software*, la vivencia de los docentes y estudiantes en ellos, identificando conocimientos, procesos de transmisión del conocimiento, medios y tecnologías, toma de decisiones, cultura organizacional y competitividad. Se continuó identificando los temas sobre modelos de calidad y gestión de procesos, administración de proyectos involucrados en el desarrollo de *software* que se

deben incluir en la gestión del conocimiento, y finalmente se construyeron matrices, se estructuró el repositorio de conocimiento y se determinó lo correspondiente a usuarios, perfiles y el plan de utilización y actualización del mismo.

Palabras claves: TSP¹, PMBOK®², SWEBOK®³, gestión del conocimiento, repositorio.

Abstract

This article highlights the problems experienced at the Autonomía University of Manizales, specifically in the Engineering program problematic systems to not having a knowledge management process for software development processes that are carried out in subjects like Business Practice, Peace and Competitiveness Research Process I and II, Design of Algorithms, Object Oriented Programming (OOP), Professional Emphasis, Technology Project Management, Software Engineering I, Software Engineering II, Software Development Project, and Bases Data of that academic program. These subjects are those that involve a higher proportion of software development. It is also perceived as problematic ignorance within the academic program, on what elements of knowledge management must remain when picking a teacher or link to it, in order to ensure the required knowledge specifically, (those of teaching) or to generate a process of training a new teacher on the topics related to software development processes.

The investigation began with an assessment of the current state of knowledge management for software development processes, the experience of teachers and students in these processes by identifying knowledge, knowledge transfer processes, media and technology, making decisions, culture and organizational competitiveness; after that project continued identifying topics about quality and process management models, and project management involved in the development of software to be included in the knowledge management process and ends with the construction of matrices, the structure of the knowledge repository, users, profiles, plan for the use and update of it.

Keywords: TSP, PMBOK®, SWEBOK®, *knowledge management, repository.*

Introducción

El Programa de Ingeniería de Sistemas de la Facultad de Ingeniería de la UAM no es ajeno al entorno altamente cambiante, en el que las tecnologías del *software* caducan rápidamente y requieren personas formadas en temas relacionados con el uso y desarrollo de *software*, que les permita afrontar estos cambios sin dejar que las empresas de las cuales ellos forman o formarán parte pierdan sus ventajas de ingreso y competitividad. Este problema hace que el programa busque de manera permanente cómo afrontar el

desarrollo de *software* y su enseñanza-aprendizaje para generar opciones de formación de sus estudiantes en competencias que les permitan desafiar estas dificultades de manera eficiente y con calidad, en un entorno en el que no sólo cambian las tecnologías de desarrollo de *software* sino que también se presenta la movilidad de los profesores a cargo de las asignaturas relacionadas con el área.

Con este trabajo se buscó generar un proceso de gestión del conocimiento en el desarrollo de *software* que se lleva a cabo en las asignaturas de Práctica Empresarial,

¹ TSP: Team Software Process. Proceso en Equipo de Software.

² PMBOK®: Project Management Body of Knowledge. Guía de los Fundamentos de la Dirección de Proyectos

³ SWEBOK®: Knowledge Management in Software Engineering. Guía del Cuerpo del Conocimiento de la Ingeniería del Software.

Paz y Competitividad⁴, Proceso de Investigación I y II, Diseño de Algoritmos, POO, Énfasis Profesional⁵, Gerencia de Proyectos Tecnológicos, Ingeniería de Software I y II, Proyecto de Desarrollo de Software, y Bases de Datos que involucran desarrollo de *software* en el programa de Ingeniería de Sistemas de la UAM. Este proceso debe incorporar de manera sistemática y formal los elementos de los modelos de calidad de *software* (modelos de proceso y de gestión) que ayudan a desarrollar *software* de buena calidad y permiten la reflexión de profesores y estudiantes en la utilización de los mismos, de manera que puedan determinar qué elementos se manejan de estos modelos, en qué casos se utilizan, y dónde están almacenados; es decir, el conocimiento en desarrollo de *software* según las exigencias del entorno, los repositorios de ese conocimiento, y su incorporación real en las asignaturas correspondientes, para facilitar la labor de los docentes y del Programa de Ingeniería de Sistemas.

De igual manera se buscó definir los elementos de gestión del conocimiento que deben estar presentes cuando se retira un docente (para conservar el conocimiento pertinente) o se vincula uno nuevo, para asegurar los conocimientos requeridos de forma específica y posiblemente capacitarlo en los temas relacionados con los procesos de desarrollo de *software* que se llevan a cabo en las asignaturas que lo involucren.

Con el fin de obtener una visión más general en este proceso, se analizaron los proyectos que trabaja la Red Colombiana de Calidad de Software (RCCS) enmarcados en los que apoyan el fortalecimiento de la capacidad en calidad de *software*, así como elementos propios derivados de investigaciones previas en la línea de calidad y métricas de software del Grupo de Ingeniería de Software y buenas prácticas de servicios y gestión de tecnologías de información en asignaturas tales como Proyecto de Desarrollo de Software, Ingeniería de Software I y II, Proceso de Investigación (cuando involucre un desarrollo de *software*), Práctica Empresarial y otras en las que se lleven a cabo proyectos de desarrollo de *software*.

La gestión del conocimiento se entiende como la instancia de administración mediante la cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar el desarrollo del conocimiento dentro de la organización o de un proyecto. Éstos son sus objetivos:

- Formular una estrategia para la organización, el desarrollo, la adquisición y aplicación del conocimiento.
- Implantar estrategias orientadas al conocimiento.
- Promover la mejora continua de los procesos del negocio, haciendo énfasis en la generación y utilización del conocimiento.
- Monitorear y evaluar los logros obtenidos mediante la aplicación del conocimiento.
- Reducir los tiempos utilizados en el desarrollo de nuevos productos, mejorar los existentes y reducir el tiempo requerido para implementar soluciones a los problemas.
- Reducir los costos generados por la repetición de errores.
- En el caso de la UAM, y el Programa de Ingeniería de Sistemas, el área en que se trabaja esta temática es la gestión del conocimiento de la ingeniería de *software*. Se parte de las siguientes premisas que han sido validadas por la comunidad científica, como lo afirma Gracia (2005):

La ingeniería de software se ocupa de teorías, métodos y herramientas necesarias para desarrollar *software* y no de materiales regidos por leyes físicas o por procesos de manufactura. El conocimiento está implícito todo el tiempo en la naturaleza de los procesos y resultados, debido a que el *software* está compuesto, de manera simplificada, de ideas plasmadas en código: es intangible. La gestión de conocimiento en la ingeniería de *software* es necesaria por la naturaleza del proceso mismo. El desarrollo de *software* es un proceso cognitivo que depende fuertemente del conocimiento y por esto mismo es indispensable gestionarlo. El proceso de desarrollo de *software* necesita conocimiento de diferentes dominios; el de la computación y el de dominio de aplicación.

⁴ Paz y competitividad: estrategia diseñada por la UAM en el cumplimiento de su misión. Es la vivencia que realizan los estudiantes de todos los programas académicos en los municipios de la ecorregión cafetera en grupos interdisciplinarios, para cooperar en proyectos de desarrollo de las alcaldías, hospitales y empresas de servicios públicos durante un semestre.

⁵ Énfasis profesional: curso sobre marcos de trabajo para la creación de aplicaciones empresariales que posean las características modernas basadas en web.

El objetivo del proceso de desarrollo es dinámico, y no está completamente definido al comienzo. El proceso de desarrollo requiere una mezcla continua entre las mentes humanas y la construcción que se va logrando (este es un proceso cíclico), y se distribuye en un grupo social y a lo largo del tiempo. No existen fronteras geográficas para ello, el conocimiento está distribuido en todo el grupo, ya que cada desarrollador aporta un poco al grupo para complementar el conocimiento. En otras palabras, la gestión del conocimiento para los desarrollos de *software* debe ser el apoyo para los grupos de trabajo en cuanto a brindar espacios que les permitan plantear un proceso de desarrollo desde el inicio hasta el final; de una manera fácil y eficiente se encuentre la información que se necesita para el desarrollo de un proyecto; se pueda llevar a cabo un seguimiento de los objetivos planteados para el mismo, y también registrar los cambios que han surgido y las implicaciones que producen; también permite a cada desarrollador retroalimentar y construir sobre lo que se ha hecho y sobre los aportes de otros desarrolladores o personas del grupo, que constituyen un apoyo que complementa los conocimientos de los integrantes del equipo para un proyecto de *software*.

Desde esta perspectiva, se ve la universidad como una empresa cuyo principal valor se centra en sus competencias esenciales, es decir, el capital intelectual. Existen dos tipos de competencias, en la primera de ellas se desarrolla el capital intelectual de operaciones y en la segunda el capital intelectual de innovación. Las operaciones de la universidad se organizan por medio de las unidades de negocios, en cambio los activos de innovación se organizan por proyectos. Para llevar a cabo estas actividades se utilizan las tecnologías de la información y las comunicaciones (TIC), las cuales permiten transferir, analizar, compartir y almacenar los conocimientos explícitos o las informaciones.

La UAM debe buscar y analizar el capital intelectual de innovación e identificar las competencias y las capacidades esenciales, reconociendo la importancia de éstas en el desarrollo de los proyectos de innovación y en la creación de conocimientos, en pro de generar ventajas competitivas sostenibles.

Igualmente, se debe considerar que la función principal de la UAM se enmarca en formación, investigación y extensión, y es allí donde se construye principalmente conocimiento, el cual se refleja así: en formación, porque de manera permanente, mediante los profesores, se está transfiriendo conocimiento a los estudiantes de la Universidad. La investigación se produce de dos maneras, cuando se genera nuevo conocimiento, en las actividades de investigación propiamente dicha, realizadas por los grupos de la institución, y también cuando hay adaptación de conocimientos, es decir, se aplica conocimiento validado por las comunidades científicas a la solución de nuevos problemas, en la investigación formativa que se procura en los estudiantes de pregrado. Se habla de extensión cuando el conocimiento que posee la universidad se utiliza para analizar y solucionar problemas del entorno o cuando éste es divulgado o transferido al medio gracias a los resultados de la investigación o de los proyectos de desarrollo.

La UAM cuenta hoy con el Programa de Ingeniería de Sistemas, el cual debe ir adaptándose a los cambios y normas que exige el mercado. Por ello, de manera permanente, se compara con otros del país y del exterior (ACM currícula) (Dewar, 2009) con el fin de ajustar su plan curricular a lo exigido por el gobierno y por el mercado, siempre buscando que los egresados obtengan competencias en el desarrollo de *software* con calidad.

Con base en los resultados reportados en los proyectos de grado, programa de Paz y Competitividad y prácticas empresariales, en los cuales la autora Beatriz Ayala ha participado como docente, supervisora y asesora durante doce años, se puede afirmar que algunas de las empresas de Manizales no utilizan metodologías ni modelos para llevar a cabo los desarrollos de *software* y no tienen un proceso definido de gestión del conocimiento en esta área dentro de sus organizaciones. Así mismo se detectó que a pesar de incluirse en el currículo asignaturas y proyectos que involucran el desarrollo de *software* a partir del uso de metodologías, modelos y técnicas, los estudiantes de pregrado y, en consecuencia, algunos egresados, no las utilizan ni las aplican en las organizaciones en las que ejercen como profesionales o practicantes.

En el proceso de desarrollo de *software* se presentan problemas tales como: dificultad de los desarrolladores para adaptarse a las herramientas y plataformas de tecnología; insuficiencia en el tiempo dedicado a obtener requerimientos del usuario; incumplimiento de los parámetros solicitados por los clientes, por lo cual se sobrepasan los tiempos de entrega de los proyectos; ausencia de adecuadas herramientas de soporte tecnológico para documentar los requerimientos de *software*, lo cual incrementa los tiempos y costos de los proyectos; contradicciones y ambigüedades frecuentes en la adquisición de requerimientos, lo que atenta contra el correcto comienzo de la vida del *software*; situaciones en las que es escaso el conocimiento sobre el dominio de la aplicación, que puede ser complejo; carencia de suficiente tiempo para obtener buenas habilidades de programación y de administración de proyectos, pues el desarrollador tarda en resolver un problema y la solución no se comparte con los compañeros, lo cual lleva a que si se presenta de nuevo y es otro desarrollador quien está a cargo, debe comenzar de cero porque pocas veces se documentan los errores. Falta formalizar las decisiones que se toman en la etapa de diseño y generar relaciones con otros (diagramas de diseño, diagramas de clase, casos de uso, diagramas de secuencia, entre otros).

La universidad también evidencia problemas presentados por los estudiantes al desarrollar su práctica empresarial, o cuando tienen su vivencia en el proceso de Paz y Competitividad, o al realizar su proceso de investigación final, cuando deben desarrollar el *software*. Estos problemas no se registran y no se cuenta con un lugar para almacenar las realizaciones después de un proceso de aprendizaje, asesorar, acompañar y mostrar algunas soluciones de los estudiantes. No se aprovechan todos los esfuerzos realizados por estos actores. En este sentido, se han identificado algunos problemas:

1. Desarticulación entre los enfoques de calidad a escala organizacional y las necesidades de mejoramiento del proyecto.
2. Rigidez de los procesos definidos para los proyectos, además de que no se adaptan al proyecto específico.
3. Falta de aplicación de buenas prácticas: no se aplican prácticas que apoyen las tareas de ingeniería (requisitos, análisis, diseño, construcción, implantación, pruebas y mantenimiento) o las tareas de soporte a la gestión de calidad (manejo de configuraciones, verificaciones, etc.). En algunos casos se realizan de manera individual, de acuerdo con las habilidades del desarrollador, y dejan a la deriva el éxito o fracaso del proyecto.
4. Dificultad para adoptar nuevas aproximaciones de desarrollo.
5. Ciclo de vida no apropiado al tipo de proyecto.
6. Software desarrollado sin las pruebas requeridas.
7. Falta de métricas que midan la calidad en productos intermedios.
8. Carencia de herramientas automatizadas e integradas.
9. Arquitectura parcialmente elaborada.
10. Falta de claridad en los papeles que se deben asumir para un proyecto.
11. Tiempos y costos altos en el desarrollo de un proyecto de *software*.
12. Inexistencia de una base de errores para reducir el trabajo.
13. Desaprovechamiento del conocimiento obtenido en proyectos previos para aplicarlo en trabajos futuros y no repetir los procesos de análisis y diseño.
14. Equipos de desarrollo que no se benefician de la experiencia existente y repiten errores.
15. Falta de claridad en los roles de quienes desarrollan *software*, por lo cual cada persona toma decisiones técnicas o de gestión.
16. Toma de decisiones basadas en el conocimiento y la experiencia personal o en el conocimiento ganado, usando contactos informales.
17. Inexistencia de procesos para compartir conocimiento de manera que los integrantes de un proyecto puedan tomar mejores decisiones.
18. Falta de preparación para la aparición de nuevas tecnologías utilizadas en el desarrollo de *software*.
19. Demoras cuando los desarrolladores y los líderes de proyecto usan una tecnología que no es familiar para el equipo de proyecto.
20. Falta de capacitación para el equipo de trabajo.
21. Necesidad de que los nuevos desarrolladores de la organización tengan conocimiento acerca del *software* existente y las convenciones de programación locales.
22. Ausencia de una estrategia para prevenir que el conocimiento de los desarrolladores desaparezca.

23. Desconocimiento de las personas de la organización que tienen el saber requerido para desarrollar un proyecto.
24. Falta de colaboración, comunicación y coordinación con los integrantes de los grupos de trabajo de desarrollo de *software* en diferentes lugares geográficos.

No existen unos lineamientos en el tema de calidad y desarrollo de *software* que definan un conjunto de objetivos y planes de acción para actualizar los programas de educación superior relacionados con esta temática, desde la perspectiva de la gestión del conocimiento.

Para la enseñanza de Ingeniería de Software en la UAM, se utiliza la estrategia de desarrollo de proyectos, que le permiten al estudiante experimentar cómo se aplican los conceptos teóricos en la práctica profesional. Sin embargo, la universidad no cuenta con una colección integrada de conocimiento. Para facilitar la operación se requiere abarcar conocimiento interno y externo, y tácito y explícito. Hasta ahora se cuenta con algunos sistemas que brindan información, como la biblioteca, mediante la cual se tiene acceso establecido a las fuentes públicas del conocimiento por medio de la comunidad académica. La red internet es un recurso que brinda a los investigadores, docentes y estudiantes acceso al conocimiento público con la ventaja de incluir documentos electrónicos. En las páginas de los grupos de investigación se encuentran los proyectos y publicaciones de los miembros del grupo; por su parte, la intranet apoya la comunicación interna por correo electrónico, y permite el acceso a las bases de datos y a los documentos electrónicos. Moodle⁶, como plataforma LMS (*Learning Management System*), sistema de gestión de cursos, alberga las aulas virtuales con los contenidos y materiales de algunas asignaturas, como también algunos vínculos a *software* de dominio público, formatos y guías de proyectos. Así mismo, se encuentra el Dot Project⁷ para registrar los proyectos cuando así lo exijan los docentes. También existe fuentes de información seleccionadas en la biblioteca de la UAM (bases de datos).

⁶ Moodle: plataforma de administración de aprendizaje LMS diseñada para proporcionarle a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados

⁷ DotProject: herramienta para la gestión de proyectos.

De acuerdo con la revista *Communications of the ACM* (Dewar, 2009), en los últimos años se ha presentado una tendencia mundial a “simplificar” los currículos con el fin de atraer alumnos, disminuyendo casi en un (1) año el tiempo de preparación como profesional. Esto ha traído como consecuencia que algunos de los pregrados en el país carecen de formación en ingeniería de software y no están preparando a los estudiantes para el desarrollo de *software* complejo y con calidad.

Por otro lado, los egresados de la universidad reflejan la problemática que se da cuando se desempeñan como profesionales del área, pues se enmarcan en actividades muy operativas, sin utilizar técnicas, ni metodologías o herramientas de ingeniería de *software* y calidad de *software*. Tampoco usan PSP (Humphrey, 1997), TSP (Humphrey, 2006), gestión de la configuración del *software*, gestión de la ingeniería de *software*, o el proceso de ingeniería de *software*; no trabajan en áreas estratégicas o tácticas e incluso limitan su participación en la creación de empresas o en proyectos de investigación.

Así mismo, ha habido un acercamiento con empresarios de la región, a los cuales se les ha solicitado su concepto sobre las falencias que pueden tener los egresados del programa de Ingeniería de Sistemas de la UAM, y han manifestado la necesidad de formar personas para liderar proyectos y desempeñarse en actividades relacionadas con la ingeniería de *software* no solamente en codificación de programas.

El Programa de Ingeniería de Sistemas de la UAM, para generar competencias en sus estudiantes, cuenta con un grupo de docentes con mucha trayectoria, pero no existe una documentación del conocimiento de sus asignaturas. Éste se pierde cuando los docentes migran hacia otras instituciones y no hay otros con el mismo conocimiento dentro del programa. No existen políticas para ir generando una base de conocimiento (medio fundamental para transferir, analizar, compartir y almacenar los conocimientos explícitos o las informaciones), analizar el capital intelectual de innovación e identificar las competencias y capacidades esenciales, enmarcando la importancia de éstas en el desarrollo de los proyectos de innovación y en la creación de nuevos conocimientos y recomendaciones al currículo, y en varios casos el contenido del aula virtual.

El proceso requerido para el desarrollo de *software* está soportado por varias asignaturas, cuyos docentes, a pesar de tener un programa para dictar dicha materia, no siempre obtienen el mismo resultado, de forma que las prácticas que los estudiantes adquieren no son las mismas y dependiendo del profesor se terminan generando competencias que no necesariamente son las que busca el programa.

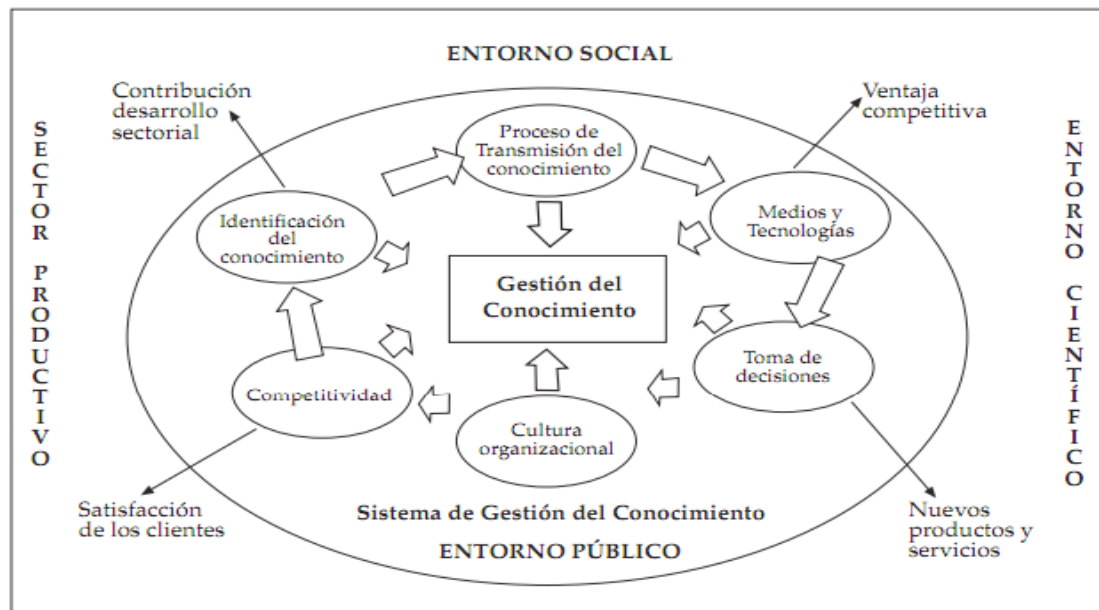
Metodología

Para el desarrollo de este trabajo se realizó un diagnóstico de la situación actual de la gestión

del conocimiento en los procesos de desarrollo de *software* que se llevan a cabo en las asignaturas del programa de Ingeniería de Sistemas de la UAM y en las vivencias de los docentes y estudiantes en este proceso. La metodología de diagnóstico empleada fue tomada de Nonaka, I. & Takeuchi (1995) y contempla los siguientes pasos:

- Identificación del conocimiento
- Procesos de transmisión del conocimiento
- Medios y tecnologías
- Toma de decisiones
- Cultura organizacional
- Competitividad

Figura 1. Modelo gráfico de la metodología para diagnosticar el estado actual de la gestión del conocimiento.



Fuente: Diseño basado en Nonaka (Nonaka, I & Takeuchi, 1995) y en el modelo de Gestión de Conocimiento de Vásquez (2001) y Piedrahita (2003, p. 87) y Gonzalez, Castro & Roncallo (2004).

Posteriormente se estudiaron los modelos de calidad de procesos, gestión de procesos y administración de proyectos, involucrados en el desarrollo de *software*. CMMI-DEV, (SEI, 2010), MPECS⁸, PSP⁹ (Humphrey, 2005), TSP, PMBOK® (Stackpole, 2010) y SWEBOK® (Abran & Moore, 2004), es decir, el conocimiento

sobre ingeniería del *software* que se incluiría en el repositorio que lo almacenará.

Como otro paso se elaboró una matriz con los tipos de objetivos que están consignadas en los programas de las diferentes asignaturas del Programa de Ingeniería de Sistemas relacionadas con los procesos de desarrollo de *software*.

También se elaboraron matrices en las cuales se identificaron para cada una de las asignaturas de los

⁸ MPECS: Mejora de Procesos Educativos en Calidad de Software (Llamosa, 2009).

⁹ PSP: Personal Software Process. Proceso Personal de Software.

procesos de desarrollo de *software* que se llevan a cabo en el Programa de Ingeniería de Sistemas de la UAM los temas que se orientan en cada una de ellas y están contemplados en MPECS, PMBOK® (Olalde, 2004), SWEBOK®, TSP y PSP (Humphrey, 1995).

Con base en el diagnóstico inicial de la gestión del conocimiento en los procesos de desarrollo de *software* en el Programa de Ingeniería de Sistemas, en las matrices de relación que se generaron en este estudio, en reuniones con jefes de práctica empresarial, y en los aspectos que se estudiaron en el marco teórico y en las variables definidas para ello, se planteó la estructura para el repositorio de conocimiento.

Como parte final se identificaron los usuarios y los perfiles de los usuarios del repositorio y así mismo se estableció un plan para la utilización y actualización del mismo y el acceso a éste según el perfil de usuario.

Este trabajo se llevó a cabo con la participación de egresados, gerentes de empresas de desarrollo de *software*, profesores y docentes de diferentes departamentos de la UAM® y externos a la institución.

Resultados

Se construyeron dos instrumentos de diagnóstico. Uno se le aplicó a ocho docentes del Programa de Ingeniería de Sistemas que tienen a su cargo 21 asignaturas relacionadas con el proceso de desarrollo de *software*, y el otro a estudiantes de Práctica Empresarial y del Programa de Paz y Competitividad. Así mismo, se generaron las tabulaciones de los diagnósticos de docentes y estudiantes de gestión del conocimiento acerca de los procesos de desarrollo de *software* en las asignaturas del Programa de Ingeniería de Sistemas de la UAM y las gráficas con las explicaciones para cada una de ellas.

Con base en MPECS, PMBOK® (2008), SWEBOK®, TSP, PSP, en los programas de las asignaturas del Programa de Ingeniería de Sistemas de la UAM, en las entrevistas con los docentes, en el diagnóstico inicial y en las entrevistas con estudiantes, se construyeron las siguientes matrices:

- Matriz de cruce entre los cursos del Programa de Ingeniería de Sistemas relacionados con el desarrollo de *software* y la guía del cuerpo de conocimiento de la Ingeniería de Software (SWEBOK®).
- Matriz de cruce entre los cursos del Programa de Ingeniería de Sistemas relacionadas con los (42) procesos de la guía del PMBOK®.
- Matriz de cruce entre los cursos del Programa de Ingeniería de Sistemas relacionadas con el proceso personal de Software (PSP).
- Matriz de cruce entre los cursos del Programa de Ingeniería de Sistemas relacionadas con el equipo de proceso de Software (TSP).
- Matriz de cruce entre los cursos del Programa de Ingeniería de Sistemas relacionadas con el desarrollo de *software* y modelo para la Mejora de Programas Educativos en Calidad de Software (MPECS).
- Roles para el Manejo de Proyectos en Mejora de Programas Educativos en Calidad de Software (MPECS) (rol de administración del proyecto y roles para el desarrollo de actividades en un proyecto).

Selección de temáticas de MPECS, PMBOK®, SWEBOK®, TSP, PSP por incluir en el repositorio; listado de cursos, mapas conceptuales y programas de las asignaturas de Ingeniería de Sistemas de la UAM; la matriz de cruce de los cursos del Programa de Ingeniería de Sistemas y sus objetivos (de aprendizaje, conceptuales, procedimentales e integradores).

La estructura del repositorio del Programa de Ingeniería de Sistemas de la UAM contempla dos partes, la primera es el lugar en el que se puede identificar, agrupar, ordenar y compartir continuamente conocimiento de todo tipo para satisfacer necesidades presentes y futuras, también definir y explotar recursos de conocimiento tanto existente como adquirido, y generar nuevas oportunidades en el desarrollo de proyectos de ingeniería de *software*. Igualmente permite conocer las temáticas expuestas en cada uno de los cursos, capacitaciones, proyectos de investigación, perfiles de los docentes y algunos elementos requeridos para el trabajo en el área de ingeniería de sistemas. En esta parte se ubican los activos de conocimiento del personal que forma

parte del programa y de los grupos de investigación vinculados a él; también contiene un espacio de gestión de conocimiento docente, conformado por la información que fue aprobada por el Departamento de Ingeniería, el Programa de Ingeniería de Sistemas, en reuniones con el personal que lo conforma o que está involucrado con los grupos de investigación de la UAM, o el personal de direccionamiento de la universidad, y un glosario en el que se incluyen aquellos términos poco conocidos, de difícil interpretación o que no sean comúnmente utilizados en el contexto en que aparecen. Cada uno tiene asociada su respectiva definición o explicación.

La segunda parte del repositorio está conformada por la sección “Activos de conocimiento”, en la que se encuentra la información que los estudiantes adicionan durante el semestre, según su vivencia en los cursos, en los proyectos y en las prácticas relacionadas con desarrollo de *software*; en otras secciones se encuentra la información de apoyo a las temáticas que se adicionan después de ser aprobadas y las verifica el comité designado para esta labor. Esta segunda parte es el lugar para identificar, agrupar, ordenar y compartir continuamente conocimiento relacionado con el objeto de esta investigación, con el fin de satisfacer necesidades presentes y futuras, e identificar y explotar recursos de conocimiento tanto existente como adquirido, y crear nuevas oportunidades en el desarrollo de proyectos de ingeniería de *software*. Se cuenta con temas de gestión del conocimiento, guía del cuerpo de conocimiento de la ingeniería de *software* SWEBOK®, dirección de proyecto PMBOK®; metodologías, activos de conocimiento suministrado por los estudiantes, tutoriales, manuales, formatos, diagramas y plantillas PSP y PMBOK®, normas, lineamientos generales y técnicas, buenas prácticas, bibliografía, documentación de errores frecuentes en los desarrollos de *software* y glosario.

Se definieron los usuarios del repositorio: docentes o investigadores, estudiantes, egresados, coordinador de programa, coordinador del departamento académico y administrador. También se creó la política del repositorio y el proceso de adquisición y transferencia del conocimiento para éste, en el cual se indicaron algunas propuestas de actualización como fueron el

sistema de recompensas: salón de la fama, sistema regalo y sistema de puntos.

También se estableció una propuesta para la elaboración de proyectos en el Programa de Ingeniería de Sistemas de la UAM, que permite formalizar y adicionar en el repositorio proyectos, de manera que incrementen la información del mismo.

Discusión

La investigación descrita en este artículo aportó a la comunidad académica de la UAM y de la región un repositorio que almacena los conocimientos de expertos y novatos sobre el desarrollo de *software* visto desde la perspectiva de la gestión del conocimiento. Lleva a formalizar y reutilizar procesos de desarrollo de *software* que de otra manera se perderían. Este repositorio no solamente sirve para almacenamiento sino como medio para la enseñanza de las estrategias que utilizan los desarrolladores en sus prácticas relacionadas con la ingeniería del *software*. La implementación del repositorio en la plataforma Moodle permite verificar que los elementos que se definieron para el proceso de gestión del conocimiento en los proyectos de desarrollo de *software*, fueran los indicados para las expectativas de la institución. Sin embargo, esta plataforma no es la más apropiada para la implementación del repositorio debido a la cantidad de usuarios, el volumen de información y las herramientas de *software* que en muchos casos requieren compatibilidad en la gestión del conocimiento. Por ello se recomienda evaluar otros tipos de plataformas que solucionen eficientemente las falencias mencionadas.

Conclusiones

Con base en la construcción de las matrices de relación entre asignaturas y los demás elementos que integran las matrices, se pueden tomar decisiones con respecto al currículo: identificar los temas correspondientes a cada asignatura y cuáles de ellos se repiten con el fin de eliminar la duplicidad temática, actualizarlos según las necesidades del contexto, y así mismo redireccionar en la

malla curricular los temas según la prioridad, los prerrequisitos y los correquisitos.

La generación del repositorio logró beneficiar el Programa de Ingeniería de Sistemas y las empresas en las que los estudiantes realizan sus procesos de práctica, proyectos y Paz y Competitividad, ya que éste se actualiza constantemente con la información de cada semestre académico.

En la implementación de los proyectos de *software*, se genera una disminución en el tiempo invertido por los desarrolladores, debido a que en el repositorio se encuentra código elaborado por otros desarrolladores, modelos reutilizables para la generación de nuevos proyectos a partir de los existentes, metodologías, tutoriales, bases de datos con registro de errores originados en los desarrollos y soluciones a los mismos, instrumentos que permiten la formalización y registros de los procesos.

Este producto da origen a un espacio para almacenar el perfil profesional de los docentes, los cursos y documentos producto del trabajo llevado a cabo en reuniones, capacitaciones, procesos de enseñanza y bibliografía que ayudan a direccionar el Proyecto Educativo del Programa (PEP).

Este trabajo integra los tipos de proyectos de gestión del conocimiento como: capturar y reusar conocimiento estructurado, capturar y compartir lecciones aprendidas desde la práctica, identificar fuentes y redes de experiencia, estructurar y mapear las necesidades de conocimiento para mejorar el rendimiento, medir y manejar el valor económico del conocimiento, sintetizar y compartir conocimiento desde fuentes externas; permite canalizar la información de diversas formas para alimentar el repositorio y al mismo tiempo ofrece el conocimiento para los usuarios de éste según sus necesidades, convierte el contenido del mismo en información más dinámica que si se utilizara sólo uno de los tipos de proyectos. Sin embargo, es importante aclarar que si las necesidades de los usuarios se enfocan a utilizar uno, dos o más de los tipos descritos, este trabajo de investigación lo permite.

El estudio de los antecedentes y la teoría sobre los modelos no es suficiente a la hora de diseñar el repositorio. Es necesario el análisis del contexto, incluyendo en él las necesidades de los usuarios, las políticas de la institución (Proyecto Educativo Institucional, PEI), el análisis del currículo (Proyecto Educativo del Programa, PEP), LAS políticas nacionales de educación, los conocimientos de los expertos y la experiencia recopilada tanto por los usuarios estudiantes como por los docentes.

Con referencia al análisis teórico que se estudió y se integró al repositorio, se puede concluir que éste forma una guía de conocimiento que apoya los procesos de enseñanza y se constituye en un recurso para las prácticas propias de la ingeniería de *software*.

La puesta en práctica de la política del repositorio permite la actualización de éste según las necesidades de los usuarios, al definir los contenidos, los metadatos, los formatos, entre otros, que ayudan a recuperar el aprendizaje adquirido en los procesos de conocimiento para lograr la gestión del mismo.

Las estrategias de motivación propuestas para la adquisición y transferencia de conocimiento logran que los saberes del docente sean almacenados y representados con la finalidad de ser aplicados a situaciones similares en escenarios de enseñanza, aprendizaje y prácticas futuras. Los estudiantes con estas estrategias recogen la teoría referente a desarrollos de *software* y la convierten en lecciones aprendidas que se pueden transferir a otros pares.

Agradecimientos

La ingeniera Beatriz Ayala Hoyos les agradece a los docentes del Departamento de Ciencias Computacionales, los estudiantes de Paz y Competitividad y de Práctica Empresarial de Ingeniería de Sistemas de la Universidad Autónoma de Manizales, quienes le brindaron información fundamental para llevar a cabo este trabajo. Así mismo, a su director de tesis de la Maestría en Gestión y Desarrollo de Proyectos de Software de la UAM, quien le brindó los conocimientos y el ánimo para terminar satisfactoriamente este trabajo.

Referencias

- Abran, A., Moore, J. W. (2004). Guide to the software engineering. Body of knowledge. SWEBOK®. A project of the IEEE Computer Society, Professional Practices Committee. Printed in the United States of America. Recuperado el 9 de julio de 2011 de <http://www.computer.org/portal/web/swebok>.
- CMMI Product Team, CMMI-DEV, V1.2.(2006). Recuperado el 28 de agosto de 2010 de <http://www.sei.cmu.edu/reports/06tr008.pdf>.
- Dewar, R. (2009). CS Education in the U.S. Heading in the wrong direction? En: *Communications of the ACM*. Vol. 52 (07).
- Gracia, J. (2005). CMM-CMMI. Recuperado el 16 de noviembre 2010 de <http://www.ingenierosoftware.com/calidad/cmm-cmmi.php>.
- González Ariza, A. L., Castro, J. P. & Roncallo, M. (2004). Diagnóstico de la gestión del conocimiento en una empresa grande de Barranquilla (Colombia). *Ingeniería & Desarrollo*. Universidad del Norte.
- Humphrey, W.S. (1997). *Introduction to the personal software process*. Reading, MA.: Addison Wesley.
- Humphrey, W. S. (2006). TSP-leading a development team, Editor Addison- Wesley.
- Humphrey, W. S. (2005). *A self-improvement process for software engineers*. Pearson Education.
- Humphrey, W.S. (1995). *A discipline for software engineering*. Reading, MA.: Addison-Wesley.
- Llamosa Villalba, R. (2009). Unión Temporal Red Colombiana de Calidad de Software (RCCS), Parquesoft, CidliS, Procesix, Universidad Eafit, Colciencias y SENA. Proyecto de Apoyo al Fortalecimiento de la Capacidad Nacional en Calidad de Software. Red Colombiana de Calidad de Software. Mejora de Programas Educativos en Calidad de Software (MPECS).
- Nonaka, I. & Takeuchi (1995). Knowledge and management (EE. UU), pp. 71-72.
- Olalde, K. (2004). Project management body of knowledge. *PMBOK Guide 2004*, PMI. Introducción a la Dirección de Proyectos. Resumen norma nacional americana ANSI/PMI 99-001-2004. Recuperado el 10 de julio 2011 de www.ehu.es/Degypi/PMBOK/cap1.htm.
- Piedrahita, J. (2003). Gestión del conocimiento. En XIII Aneiap. Congreso Nacional sobre Gestión Estratégica de la Información. Universidad de Medellín.
- Project Management Institute. (2008). *Guía de los fundamentos para la dirección de proyectos: Guía del PMBOK* (4.ª ed.). Pennsylvania: Project Management Institute.
- SEI, CMMI Product Team, CMMI-DEV. (2010). Recuperado el 28 de agosto de 2010 de <http://www.sei.cmu.edu/reports/06tr008.pdf>.
- Stackpole, C. (2010). *A user's manual to the PMBOK Guide*. John Wiley and Sons (ed.).

Sobre los autores

Beatriz Ayala Hoyos

Ingeniera de sistemas, especialista en Gerencia Financiera y magíster en Gestión y Desarrollo de Proyectos de Software de la Universidad Autónoma de Manizales (UAM). Profesora instructora del Departamento de Ciencias Computacionales, coordinadora de Prácticas, asesora de Paz y Competitividad de la misma institución. Manizales, Colombia. bayala@autonoma.edu.co

Mauricio Alba Castro

Ingeniero de sistemas y computación de la Universidad de los Andes, magíster universitario

en Métodos Formales, Ingeniería de Software y Sistemas de Información y Ph.D. en Informática de la Universidad Politécnica de Valencia. Profesor titular del Departamento de Ciencias Computacionales de la UAM. Manizales, Colombia. malba@autonoma.edu.co

Herminia Quiceno Valencia

Licenciada en Filosofía y Letras y magíster en Filosofía de la Universidad de Caldas. Profesora asociada del Departamento de Educación de la UAM. quiher@autonoma.edu.co

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.