

Herramientas de gestión de conocimiento utilizadas en proyectos Software

Luis Fernando Sierra Joya

Estudiante de maestría en Ingeniería Industrial
Universidad Industrial de Santander, Bucaramanga, Colombia, luis.sierra@correo.uis.edu.co

Luis Eduardo Becerra Ardila

Profesor Asociado
Universidad Industrial de Santander, Bucaramanga, Colombia, lbecerra@uis.edu.co

Luis Carlos Gómez Flórez

Profesor Titular
Universidad Industrial de Santander, Bucaramanga, Colombia, lcgomezf@uis.edu.co

RESUMEN

La gestión del conocimiento ha tomado gran importancia en la comunidad empresarial y académica en el entorno del desarrollo de proyectos software, dado que el factor humano es un componente imprescindible, el cual constituye un activo valioso, principalmente por el carácter intangible del conocimiento. El objetivo de este trabajo es mostrar herramientas usadas en proyectos de desarrollo software que favorecen la gestión de conocimiento mediante una revisión de publicaciones en el área. Al conocer este tipo de herramientas y la forma de uso pueden ser empleadas en gestionar el conocimiento de las personas que intervienen en el desarrollo del software.

Palabras claves: Gestión del conocimiento, proyectos software, gestión de proyectos software.

ABSTRACT

Knowledge management has become very important in the business and academic community in the environment of software project development, given that the human factor is an essential component, which is a valuable asset, especially by the intangible nature of knowledge. The aim of this paper is to show tools used in software development projects that promote knowledge management by reviewing publications in the area. By knowing these tools and how to use can be used to manage the knowledge of the people involved in software development.

Keywords: Knowledge management, software projects, software project management.

1. INTRODUCCIÓN

Las organizaciones en donde se desarrolla software requieren un uso intensivo de conocimientos, actividades y técnicas en las cuales participan muchas personas que trabajan en las diferentes fases y actividades del proceso de desarrollo. Estos conocimientos no son estáticos en la organización, sino que se encuentran en constante crecimiento a medida que se desarrollan diversos proyectos (Corbin, Dunbar, & Zhu, 2007). En este tipo de organizaciones el conocimiento de los individuos juega un papel muy importante y se convierte en el principal activo dejando en un segundo plano los edificios o maquinas costosas (Rus & Lindvall, 2002).

Las organizaciones empiezan a prestar mayor atención al conocimiento de las personas (activos intangibles), considerándose desde entonces el conocimiento como la principal fuente de ventaja competitiva de una organización (Nonaka & Takeuchi, 1995; Nonaka, von Krogh, & Voelpel, 2006). Se han realizado numerosas

investigaciones alrededor de la esta temática con el fin de aprovechar el conocimiento de las personas pertenecientes a la organización y hacerlo de fácil acceso. Además, surgen múltiples desarrollos de tecnologías que contribuyen a dar soporte a los procesos para gestionar conocimiento (Alavi & Leidner, 2001).

En este trabajo se mostrara una revisión de literatura en las bases de datos científicas acerca de las herramientas utilizadas en gestión del conocimiento para los proyectos software conociendo la literatura que al respecto aborda esta temática. En la siguiente sección se presenta algunos conceptos de gestión de conocimiento, desarrollo de software y gestión de proyectos, la metodología empleada y los resultados encontrados.

2. CONCEPTOS BÁSICOS

2.1.1 GESTIÓN DEL CONOCIMIENTO

Al revisar la literatura se encuentran diversas definiciones para gestión del conocimiento, para el efecto del presente trabajo se tomara un autor ampliamente referenciado el cual introduce la siguiente definición: La gestión de conocimiento es la capacidad de la empresa para crear conocimiento nuevo, diseminarlo en la organización e incorporarlo en productos, servicios y sistemas (Nonaka & Takeuchi, 1995). Al hablar de conocimiento necesariamente se deberá hablar también de información y datos, al respecto se encuentra a Davenport y Prusak (1999), quienes presentan la diferencia entre estos conceptos y se resumen en la tabla 1.

Tabla 1: Dato, Información, Conocimiento

Dato	Información	Conocimiento
<p>Observaciones sencillas de los estados del mundo.</p> <ul style="list-style-type: none"> • Se estructuran fácilmente • Se capturan con facilidad en las máquinas. • Se transfieren con facilidad 	<p>Datos dotados de pertinencia y propósito.</p> <ul style="list-style-type: none"> • Requieren una unidad de análisis • Necesita consenso sobre el significado • La intermediación humana es importante 	<p>Información valiosa de la mente humana, incluye reflexión, síntesis y contexto.</p> <ul style="list-style-type: none"> • Difícil de estructurar • Difícil de capturar en las máquinas • A menudo es tácito • La transferencia es complicada

Fuente (Davenport & Prusak, 1999)

Alavi y Leidner (2001) mencionan la similitud entre dos tipos de gestión, la gestión de conocimiento y la gestión de la información; a pesar que ambas implican el desarrollo de diferentes sistemas de gestión, la información como el conocimiento pueden someterse a procesos similares: adquisición, creación, retención, almacenamiento, transferencia y aplicación.

2.1.2 DESARROLLO DE SOFTWARE

La disciplina de la computación encargada de ofrecer métodos y técnicas para el desarrollo y mantenimiento del software de calidad es la Ingeniería del Software (ACM, AIS, & IEEE-CS., 2005; Pressman, 2010) Así mismo, la IEEE (Institute of Electrical and Electronics Engineers) Computer Society define la ingeniería de software como: “La aplicación de un enfoque sistemático, disciplinado, hacia el desarrollo, operación y mantenimiento del software; es decir la aplicación de ingeniería del software.” (IEEE Std 610.12, 1990, p.67).

Por otra parte, el proceso de desarrollo software es definido por el IEEE estándar 610.12 (1990) como un proceso que consiste en traducir las necesidades de los usuarios en los requisitos de software, la transformación de los

requisitos de software en el diseño, la codificación, pruebas y mantenimiento del software. A su vez estas actividades pueden superponerse o realizarse iterativamente, dependiendo del modelo de ciclo de vida del software seleccionado dependiendo del alcance del proyecto, de la situación problema a resolver y de la necesidad de realimentación e iteración de las etapas de desarrollo (Palma, Paniagua, Mart, & Mar, 2000), decisión que en ocasiones es un factor determinante del éxito o fracaso del proyecto software.

2.1.3 GESTIÓN DE CONOCIMIENTO EN EL DESARROLLO DE SOFTWARE

La invención de software requiere un esfuerzo amplio en el levantamiento de requerimientos, análisis, validación y diseño; estos muchas veces se ven afectados por no tener en cuenta los errores cometidos en el pasado, haciendo más costosa la implementación de estos proyectos (Corbin et al., 2007). Por otra parte, hay una alta rotación del personal en organizaciones que desarrollan software, llevándose con ellos el conocimiento adquirido durante el ejercicio de su trabajo, lo cual muestra la necesidad de un mejor aprovechamiento de los conocimientos en la organización (Torgeir Dingsøy & Conradi, 2002; Rus & Lindvall, 2002).

Algunos autores como (Aurum, Daneshgar, & Ward, 2008; Bjørnson, 2007; Corbin et al., 2007; Desouza & Awazu, 2006; Torgeir Dingsøy & Conradi, 2002; Iversen, Mathiassen, & Nielsen, 2004; Rus & Lindvall, 2002) destacan la importancia de incorporar la gestión de conocimiento en la ingeniería del software, principalmente por su la variedad y gran proporción de conocimientos. Corbin et al. (2007), destaca que la gestión de conocimiento permite en los proyectos de desarrollo de software, descubrir nuevo conocimiento, capturarlo, almacenarlo, recuperarlo, compartirlo y entenderlo.

Incorporar la gestión de conocimiento en el proceso de desarrollo software contribuye a mejorar este proceso en el sentido tal como menciona Aurum et al. (2008), la gestión de conocimiento contribuye a reducir los costos, cumplir con los tiempos planificados y mejorar la calidad del proyecto. Las organizaciones desarrolladoras de software pueden ver la gestión del conocimiento como el aprovechamiento del conocimiento de los individuos en el beneficio de la empresa permitiéndoles mejorar varios aspectos, como los siguientes señalados por Bjornson (2007): Pérdida de conocimiento al no uso continuo en el tiempo, falta de conocimiento y un tiempo demasiado largo para adquirirlo, curvas de aprendizaje muy pronunciadas, gente que repite los errores y la realización del doble de trabajo porque se olvidaron de lo que han aprendido de proyectos anteriores, y finalmente individuos que poseen un conocimiento clave que debe estar disponible para el resto de la organización.

3. METODOLOGÍA

Para el desarrollo de este estudio se realizó una revisión de literatura, el proceso seguido se expone a continuación. Inicialmente, en una fase preliminar a la revisión se hizo un análisis de las publicaciones relevantes con el fin de establecer un marco teórico base para la investigación, posteriormente se identificaron las palabras claves para realizar la revisión.

Lo siguiente fue plantear el objetivo de la revisión el cual es: analizar las publicaciones científicas para encontrar herramientas de gestión de conocimiento en proyectos de desarrollo Software. Posteriormente se realizó un análisis de las publicaciones científicas en la base de datos Scopus, dado que esta es referente científico internacional y posee artículos de calidad, luego se definió el protocolo de revisión el cual comprendió la definición de la ecuación de búsqueda para gestión de conocimiento y proyectos software, la búsqueda tomo como periodo de tiempo artículos publicados en un periodo de tiempo entre el 2001 y 2012 y la definición de los criterios de inclusión y exclusión.

Después de refinar la ecuación de búsqueda con el fin de seleccionar las publicaciones relevantes a la temática seleccionada y el filtrando de información con base en los criterios establecidos, se obtuvieron 155 publicaciones para analizar y seleccionar los artículos a leer.

4. RESULTADOS

Se empezara mostrando los resultados analíticos y posteriormente algunas herramientas encontradas de gestión de conocimiento utilizables proyectos Software.

4.1 RESULTADOS ANALÍTICOS

4.1.1 TENDENCIA POR AÑOS

En la figura 1 se observa un creciente número de publicaciones en el área de estudio, a partir del año 2007 se aprecia un aumento significativo, lo que indica un creciente interés de la comunidad científica en la temática sobre gestión de conocimiento en proyectos software. El periodo analizado entre los años 2001 al 2012 se obtuvo 155 documentos.

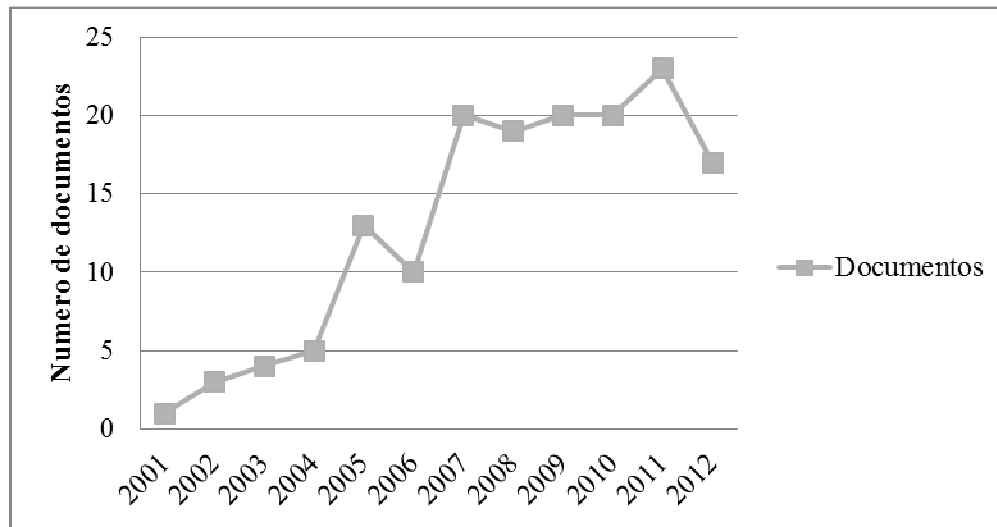


Figura 1: Publicaciones por año.

4.1.2 TIPOS DE DOCUMENTOS

En la tabla 2 se muestra los tipos de documentos encontrados, se desglosan si son artículos en revista, conferencias y revisiones. Se obtiene como resultado más artículos de tipo conferencia y agrupando en dos grupos se obtuvieron 43 artículos en revista y 112 artículos publicados en conferencias.

Tabla 2: Tipos de documentos

Tipo de Documento	No Documentos	Porcentaje
Artículo en conferencia	102	65,8
Artículo en revista	38	24,5
Artículo revisión en conferencia	10	6,5
Artículo de tipo revisión	5	3,2
Total	155	100%

4.1.3 PUBLICACIONES POR PAÍS

Se analizó el top 10 de publicaciones dentro del conjunto de países encontrados, los cuales representan el 79,35% del total de publicaciones, en primer lugar aparece Estados Unidos con un 21.29%, seguido de china con 11.61%,

en Latinoamérica se destaca en un quinto puesto Brasil con el 5.81% de publicaciones en la temática. A continuación en la tabla 3 se observa el ranking de países.

Tabla 3: Países

País	Cantidad de publicaciones	% de publicaciones
Estados Unidos	33	21,29
China	18	11,61
Alemania	17	10,97
Finlandia	10	6,45
Brasil	9	5,81
España	8	5,16
Reino Unido	8	5,16
Canada	7	4,52
Noruega	7	4,52
Grecia	6	3,87
	123	79,35

4.1.4 PUBLICACIONES EN REVISTAS Y CONFERENCIAS

Las revistas y conferencias científicas con mayor número de publicaciones se encuentran detalladas en la tabla 4, en la cual se encuentran un análisis del top 10 con el nombre de la revista o conferencia con la respectiva cantidad de artículos, en el primer y segundo lugar se tienen las revistas de lecture Notes in Computer Science con 9 publicaciones representando un 5.81% del total de publicaciones, Information and Software Technology con 7 publicaciones representando un 4.52% del total de publicaciones, en la revista Information and Software Technology se encuentran artículos muy completos y extensos que facilitan la comprensión y el estudio por lo cual se resalta esta revista ya que es una excelente fuente de consulta en la temática investigada.

Tabla 4: Revistas y conferencias

Revistas y conferencias	Cantidad
Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics	9
Information and Software Technology	7
ACM International Conference Proceeding Series	4
Proceedings International Conference on Software Engineering	4
Proceedings of the Annual Hawaii International Conference on System Sciences	4
IEEE Software	3
Journal of Systems and Software	3
Software Process Improvement and Practice	3
Icsoft 2008 Proceedings of the 3rd International Conference on Software and Data Technologies	2
Communications in Computer and Information Science	2

4.1.5 AUTORES

Los autores con más publicaciones en el área son Stamelos, y Dingsort con 5 y cuatro publicaciones respectivamente, Stamelos, Ioannis G. de la Aristotle University of Thessaloniki, Department of Informatics,

Thessaloniki en Grecia y Dingsøy, Torgeir del Sintef Foundation for Scientific and Industrial Research At the Norwegian Institute of Technology, Trondheim en el País de Noruega. En la tabla 5 se muestran los autores con más publicaciones en el área.

Tabla 5: Autores

Autor	Publicaciones
Stamelos, I.	5
Dingsoyr, T.	4
Seaman, C,B.	3
Travassos, G,H.	3
Mitchell, S,M.	3
Knauss, E.	2
Crawford, B.	2
Demetriadis, S,N.	2
Lianq, T,P.	2
Bohner, S,A.	2

4.2 HERRAMIENTAS ENCONTRADAS

El intercambio de conocimientos entre los proyectos pueden ayudar a asegurar el éxito, sin embargo, el lograr el aprendizaje de proyectos anteriores no es una tarea fácil, es más fácil decirlo que hacerlo, la reutilización del conocimiento en los proyectos de software es muy importante, pero difícil en la práctica (Petter, Mathiassen, & Vaishnavi, 2007).

Hay varios tipos de herramientas de intercambio de conocimientos a disposición de los directores de proyectos de software como se aprecia en la tabla 6:

Tabla 6: Herramientas para compartir conocimientos

Herramientas	Ejemplos de estas herramientas comerciales
Sistemas colaborativos	WebEx (http://www.webex.com)
Repositorios de contenido	Microsoft SharePoint (http://www.microsoft.com/sharepoint)
Inversión y sistemas de gestión de portafolios	Primavera project portfolio management (http://www.primavera.com)
Mapas de conocimiento	The Brain (http://www.thebrain.com)
Redes de trabajo	
Análisis postmortem	
Sistemas de gestión de proyectos	Microsoft project (http://www.microsoft.com/project) or Primavera (http://www.primavera.com)
Evaluación de riesgos	Palisade Corp. (http://www.palisade.com) or Relational Security Corp. (http://www.relsec.com)
Análisis de debilidades, fortalezas, Amenazas y	

oportunidades (matiz DOFA o FODA)	
Plantillas	Templates: Method123 (http://www.method123.com)

Fuente: Elaboración basada en (Petter et al., 2007).

Los sistemas colaborativos permiten a las personas conectarse virtualmente a reuniones, discusiones, lluvia de ideas y compartir conocimiento. Usando sistemas de colaboración los participantes crean nuevo conocimiento vía exploración y desarrollo, lo cual permite una comprensión del saber porque (*know-why*) de los proyectos (Petter et al., 2007). Este tipo de conocimiento es tácito al usar sistemas colaborativos y se centra en la creación de soluciones que se ajustan al contexto presente del proyecto. Repositorios de contenido almacenan el conocimiento documentado en una base de datos centralizada para crear un almacén de conocimientos dentro de la organización y ayudan a explotar los conocimientos existentes mediante la reutilización en futuras situaciones. La atención se centra en el almacenamiento de las mejores prácticas y métodos para documentar los conocimientos obtenidos en proyectos anteriores, con ello la captura de pericia organizativa.

Las inversiones y sistemas de gestión de portafolios se realizan para identificar, planificar y obtener un seguimiento de las inversiones en los proyectos. Esta herramienta es útil para seleccionar y planificar futuros proyectos sobre la base de las necesidades de la organización. Los líderes del proyecto pueden explotar el conocimiento de los proyectos anteriores, generados por la organización y el entorno, con el fin de desarrollar y documentar diferentes aspectos que lleven a la organización a tomar decisiones para contratar o mantener sus inversiones en un proyecto (Petter et al., 2007). Por otra parte, los mapas de conocimiento ayudan a quienes buscan algún conocimiento lo puedan encontrar en las personas que tengan la experiencia necesaria.

El Análisis post-mortem es a la vez un proceso y una serie de documentos para identificar y documentar los éxitos y fracasos de un proyecto determinado. El conocimiento obtenido mediante vía post-mortem puede ayudar en la reducción de la repetición de los errores del pasado, mediante la obtención de un recuento histórico de lo que salió mal, la mejora de los procesos y el desempeño en proyectos futuros (T. Dingsøyr, 2005); lo cual apunta a explicaciones de saber por qué del conocimiento, a través del análisis exploratorio y a menudo documentado por el administrador del proyecto o de los miembros del equipo del proyecto para facilitar el intercambio de toda la organización (Petter et al., 2007) .

Las plantillas son documentos, tales como planes de proyectos, presupuestos, o documentación de proyectos anteriores, que son continuamente reutilizados para mantener la consistencia entre los proyectos. Una parte importante del tiempo se dedica a documentar un proyecto a través de planes de desarrollo, de calidad, y prueba, así como requisito y las especificaciones de diseño. Las plantillas compartidas entre colegas pueden ayudar a los gerentes de proyectos de software aprovechar el saber cómo (*know-how*), del conocimiento en proyectos anteriores.

5. CONCLUSIONES

En la última década se hallaron gran número de publicaciones acerca de herramientas de gestión de conocimiento que pueden ser usadas en proyectos de desarrollo software, lo que muestra un gran interés de la comunidad científica en este tipo de estudios y el potencial de estudio en esta área.

En desarrollo de software hay un fuerte stock de capital intelectual de las personas, quienes son los que tienen la responsabilidad de realizar un producto intangible como es el software, estos individuos tienen una alta rotación ya sea cambiando de organización o de proyecto dentro de la misma organización, llevándose con ellos el conocimiento, fruto de la experiencia adquirida en su trabajo, ocasionando que la organización pierda ese conocimiento y sea ineficiente en el desarrollo de algunos procesos, por lo cual es importante conocer y poner en uso herramientas que contribuyan a la adecuada gestión de los conocimientos.

AGRADECIMIENTOS

Los autores agradecen a la Universidad Industrial de Santander y al Departamento Administrativo de Ciencia, Tecnología e innovación, Colciencias, por el apoyo recibido, a través del el programa de Jóvenes Investigadores e innovadores, para el desarrollo de la investigación en la cual se enmarca este trabajo.

REFERENCIAS

- ACM, AIS, and IEEE-CS. (2005). Computing Curricula. *Overview Report. Draft*. Retrieved from <http://www.acm.org/education/curricula.html> (descargado el 15 de octubre de 2011).
- Alavi, M., and Leidner, D. (2001). "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research issues". *MIS Quarterly* Vol. 25 pp.107–136.
- Aurum, A., Daneshgar, F., and Ward, J. (2008). "Investigating Knowledge Management practices in software development organisations – An Australian experience". *Information and Software Technology*, Vol. 50, pp 511-533.
- Bjornson, F.O, and Dingsoyr, T. (2007). " Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used". *Information and software technology* Vol. 50, pp 1055-1068.
- Corbin. R., Dunbar, C., and Zhu, Q. (2007). "A three-tier Knowledge Management Scheme for Software Engineering Support and Innovation". *The Journal of System and Software*, Vol. 80, pp 1494-1505.
- Davenport, T. H., Prusak, Laurence. (1997). *Ecología De La Información*, Oxford University Press.
- Desouza, K. ., and Awazu, Y. (2006). Knowledge Management at SMEs: five peculiarities. *Journal of Knowledge Management*, Vol 10 No1, pp 32–43.
- Dingsoyr. T., and Conradi, R. (2002). "A survey of case studies of the use of knowledge management in software engineering". *International Journal of software Engineering and Knowledge engineering*, Vol. 12 No. 4, pp 391-414.
- Dingsøy, T. (2005). Postmortem reviews: Purpose and approaches in software engineering. *Information and Software Technology*, Vol 47 No 5, pp 293–303.
- IEEE Std 610.12. (1990). *IEEE Standard Glossary of Software Engineering Terminology. Office* (Vol. 121990).
- Iversen, J. H., Mathiassen, L., & Nielsen, P. A. (2004). Risk in Software Process An Action Improvement : *Management Information Systems*, Vol 28 No 3, pp 395–433.
- Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company. How Japanese Companies Create the Dynamics of Innovations*. Nueva York.: Oxford University Press.
- Nonaka, I., Von Krogh, G., and Voelpel, S. (2006). Organizational knowledge creation theory: Evolutionary paths and future advances Organization Studie. *Organization Studies*, Vol 27 No 8, pp 1179–1208.
- Palma, J. T., Paniagua, E., Mart, F., and Mar, R. (2000). Ingeniería del conocimiento. de la extracción al modelado de conocimimient. *Revista Iberoamericana de Inteligencia Artificial*, 46–72.
- Petter, S., Mathiassen, L., and Vaishnavi, V. (2007). Five keys to project knowledge sharing. *IT Professional*, Vol 9 No 3, pp 42–46.
- Pressman, R. (2010). *Ingeniería de Software. Un enfoque práctico. Séptima edición*. Mc Graw Hill.
- Rus, I., and Lindvall, M. (2002). "Guest Editors' Introduction: Knowledge Management in Software Engineering". *IEEE Software*, Vol. 19, No. 3, pp. 26-38.

Autorización y Renuncia

Los autores autorizan a LACCEI para publicar el escrito en las memorias de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito.