

Flujo de Conocimiento para la Localización del Expertise en el Desarrollo de Software: Implicaciones de Diseño

José Ramón Martínez García¹, Ramón Rene Palacio Cinco²,
Luis-Felipe Rodríguez¹ y Oscar M. Rodríguez-Elias³

¹ Instituto Tecnológico de Sonora, Unidad Nainari,
Antonio Caso 2266 CP. 85130, Ciudad Obregón, Sonora, México.
joseramonmg26@gmail.com, luis.rodriguez@itson.edu.mx

² Instituto Tecnológico de Sonora, Unidad Navojoa,
Ramón Corona S/N, Col. ITSON, CP. 85860, Navojoa, Sonora, México.
ramon.palacio@itson.edu.mx

³ Instituto Tecnológico de Hermosillo
Sahuaro Indeco, CP. 83170, Hermosillo, Sonora, México.
omrodriguez@ith.mx

Resumen: El presente trabajo aborda el tema de la búsqueda de expertise en el desarrollo de software, donde se tiene como objetivo obtener información para la creación de un mapa del conocimiento, el cual facilite la identificación de los elementos de información, cómo los desarrolladores buscan el expertise en su organización, y cuáles son los obstáculos a los que se enfrentan. Para lograr dicho objetivo se utilizó la metodología KoFI para identificar las fuentes, el flujo de conocimiento y los obstáculos que se presentan en la búsqueda de expertise. Para dar una solución a estos obstáculos, se proponen mecanismos que permiten gestionar el conocimiento en organizaciones de desarrollo de software de tal forma que se tenga acceso a dicho conocimiento respecto a la información de disponibilidad de los proveedores del expertise. Como resultado se obtuvieron las implicaciones de diseño para construir una herramienta que permita la búsqueda de expertise.

Palabras clave: Búsqueda de Expertise, Gestión del Conocimiento, Desarrollo de Software.

1 Introducción

El entorno actual de las organizaciones se caracteriza por el cambio continuo y la evolución en cuanto a las prácticas y tecnologías utilizadas. Las acciones deben ser anticipadas y adaptativas, basadas en un ciclo rápido de creación de conocimiento. De ahí la necesidad de un proceso de negocio que formalice la gestión y aseguramiento de los activos intelectuales [1, 2]. Es por ello que se han hecho esfuerzos por tener una adecuada gestión del conocimiento, la cual es ampliamente conocida y practicada por grandes organizaciones, como una herramienta útil para que cada uno de los miembros de la empresa sepa lo que otros conocen para mejorar los resultados de sus actividades. De manera que si las organizaciones pueden manejar el proceso de aprendizaje, pudieran ser más eficientes [3]. Este tipo de gestión se compone de una

serie de prácticas que permiten identificar crear, representar y distribuir el conocimiento para su reutilización, distribución y aprendizaje. Por ello existe un gran interés en el tratamiento de conocimiento como un recurso significativo en las organizaciones, enfocándose en los conocimientos humanos, y cómo explotarlos para tener el máximo rendimiento en la organización [4].

El creciente interés por el conocimiento y la gestión del conocimiento organizacional se deriva de la transición a la economía del conocimiento, donde el conocimiento es visto como la principal fuente de creación de valor y ventaja competitiva sostenible [5, 6]. Por tal motivo, la gestión del conocimiento tiene como meta mitigar la pérdida y el riesgo en los procesos de producción de la organización, mejorar la eficiencia de la organización e innovar. Esto añade un gran valor a las organizaciones, de manera que se pueden tomar mejores decisiones, tener un flujo libre de ideas que llevan al conocimiento e innovación, eliminar los procesos redundantes, mejorar el servicio al cliente y la eficiencia conduciendo a una gran productividad.

La gestión del conocimiento ha beneficiado a diferentes áreas, tales como de educación [7, 8], cuidado de la salud [9, 10], desarrollo de software [11, 12], entre otras. Particularmente en la industria del software se han abordado los retos de adaptación del conocimiento a las tecnologías emergentes (p. ej. cuando los desarrolladores utilizan una tecnología que no es familiar para los miembros del equipo, ellos utilizan la técnica de “aprender sobre la marcha”, lo que muchas veces resulta en retrasos), acceso al dominio del conocimiento (p. ej. cuando una organización debe adquirir dominio de conocimientos nuevos, ya sea por formación o mediante la contratación de empleados con conocimientos y difundirla por todo el equipo), intercambio del conocimiento sobre las políticas y prácticas locales (p. ej. cuando los desarrolladores suelen difundir el conocimiento a través de reuniones informales, de manera que, no todo el mundo tiene acceso al conocimiento que necesita), capturar el conocimiento y saber qué es lo que sabe cada uno (p. ej. cuando algún desarrollador con conocimiento crítico deja la organización, crea huecos de conocimiento que ninguno se da cuenta hasta que es necesario dicho conocimiento) y colaboración e intercambio de conocimiento (p. ej. cuando los miembros de un equipo a menudo están distribuidos geográficamente con diferentes zonas horarias y deben interactuar para intercambiar o transferir información) [13].

El desarrollo de software se caracteriza por ser un actividad intelectual compleja que requiere de la interacción constante con los colaboradores de la organización. Así mismo, el desarrollo de software es considerado un proceso de constante cambio, donde muchas personas trabajan en diferentes fases, actividades y proyectos. Tales características producen los siguientes inconvenientes [13]: i) *Re-trabajo*: Se refiere a cuando un desarrollador trabaja sobre un artefacto que otro ya había realizado anteriormente para otro proyecto, y pudo haberse reutilizado, reduciendo así el tiempo para realizar dicha actividad. ii) *Fracaso en las Consultas*: Se refiere a cuando un usuario busca una solución a cierto problema en el código, y tiene dificultades para encontrar una solución en algún foro de programadores en línea, manuales o video tutoriales. iii) *Asesoría Inadecuada*: Se refiere a cuando se recibe consultoría de varias personas para solucionar un problema de código o de una actividad, y ninguna puede dar una solución adecuada al problema.

Este tipo de inconvenientes traen como consecuencia problemas de producción (p. ej. generan costos debido a los retrasos en las actividades), comunicación (p. ej. es afectada por los retrasos generando desconfianza entre los miembros del equipo), retrasos (p. ej. ocurren cuando un desarrollador depende del artefacto de un colega para poder iniciar con su actividad), clarificación (p. ej. cuando la comunicación se ve afectada por la coordinación produce retrasos) [14, 15].

Una de las maneras de abordar lo anterior es mediante la *localización de expertos*, puesto que se pretende encontrar personas con ciertas habilidades que pudieran ayudar a un colega a solucionar algún problema en particular que no le permite avanzar en su trabajo. En el caso de los desarrolladores de software, cuando tienen dificultad para realizar alguna actividad, suelen ir en busca de conocimiento, donde la meta es encontrar el *expertise* (conocimiento de mejor). Es decir, quien posee experiencia es capaz de realizar una tarea mucho mejor que los que no la tienen. Se trata de un conocimiento específico en su mejor momento. Cabe mencionar que la palabra "experto" puede ser usado para describir a las personas que poseen altos niveles de habilidades o conocimientos [16]. Por lo anterior, se definen las siguientes preguntas de investigación que guían este artículo:

- ¿Cuáles son las fuentes de conocimiento en el desarrollo de software?
- ¿Cuáles son los elementos de información clave para identificar a un experto y el *expertise*?
- ¿De qué manera está representada la información?
- ¿Cuáles son las barreras que impiden el flujo de la información?

Para responder estas preguntas se analizó el flujo del conocimiento para la búsqueda de *expertise* en el desarrollo de software, por lo que el objetivo de este artículo es obtener información para la creación de un mapa del conocimiento, el cual facilite la identificación de los elementos de información, cómo los desarrolladores buscan el *expertise* en su organización, y cuáles son los obstáculos a los que se enfrentan. Los resultados obtenidos en este trabajo son los mecanismos para dar soporte a la búsqueda de *expertise* dentro de las actividades del desarrollo de software.

2 Flujo del Conocimiento para la Búsqueda de Expertise en el Desarrollo de Software

La metodología para identificar el flujo de conocimiento para la búsqueda de *expertise* fue KoFI (Knowledge Flow Identification) [17, 18], la cual consta de cuatro fases. La Fase 1 consiste en la identificación de las diferentes fuentes en las que se genera o almacena el conocimiento; La Fase 2 permite identificar los tipos de conocimiento utilizados y generados en los procesos principales de la organización, mientras que la Fase 3 identifica cómo fluye el conocimiento dentro de la organización. Por último, la Fase 4 consiste en la identificación de los principales problemas que obstaculizan el flujo de este conocimiento. Para esto, se realizó un estudio con desarrolladores de software de varias organizaciones, con diferentes tipos de prácticas de desarrollo (centralizado, distribuido y global). La característica común de estas organizaciones es que su proceso de producción está basado en metodologías

ágiles. Participaron ocho trabajadores, incluyendo dos líderes de proyecto, cuatro desarrolladores y dos ingenieros en software. Para obtener la información de los participantes se utilizó la técnica de la entrevista semi-estructurada. La entrevista consistía en preguntas de temas relacionados a colaboración, coordinación, intercambio de conocimiento y la administración de los proyectos.

El propósito de las entrevistas fue entender cómo se realiza el proceso de búsqueda de *expertise* entre los miembros de los equipos de desarrollo de software. Se llevaron a cabo 8 entrevistas de forma individual y se guardó en formato de audio. La duración de las entrevistas fue de 40 minutos en promedio. Los datos fueron extraídos de las entrevistas usando diagramas de afinidad, que es una herramienta que sintetiza un conjunto de datos verbales (p. ej. ideas, opiniones, expresiones) agrupándolos en función de la relación que tienen entre sí [19]. Este proceso se inició con la transcripción de las entrevistas para encontrar los datos clave de las respuestas de los participantes. A partir de eso se clasificaron los datos de las respuestas que aparecían más recurrentemente. Posteriormente continuamos con el análisis de los datos para identificar las relaciones entre los procesos de búsqueda de *expertise*. Por último, a partir del diagrama de afinidad y las categorías definidas se obtuvieron conclusiones.

Con la información recopilada, en los siguientes apartados se presentan los resultados obtenidos.

2.1 Fase 1: Identificando las Fuentes del Conocimiento

En esta fase fue necesario tener en cuenta las fuentes de conocimiento que podrían ser utilizadas para localizar a un experto para dar solución a un problema dentro de una actividad. Los tipos de fuentes encontrados a partir de las entrevistas fueron libros, manuales, blogs, código reutilizado y consulta con compañeros o expertos externos. Esta información coincide con las categorías que propone [20], las cuales son:

- *Personas*: El conocimiento se almacena en las personas ya sea a nivel individual o dentro de un grupo o un conjunto de personas.
- *Artefactos*: El conocimiento se encuentra en las prácticas de la empresa como las rutinas diarias de trabajo, tecnologías o repositorios de documentos físicos o digitales (p. ej. libros, manuales, videos).
- *Entidades Organizacionales*: El conocimiento en las entidades de organización como unidades organizativas (partes de la organización), como la organización entera, y en las relaciones entre organizaciones (p. ej. relación entre sus clientes).

2.2 Fase 2: Identificando los Tópicos del Conocimiento

En esta fase fue necesario identificar los temas de conocimiento implicados en el proceso de búsqueda de *expertise* en el desarrollo de software, teniendo en cuenta los diferentes tipos de conocimientos generados por la organización. Los tipos de conocimiento relevantes para este trabajo están relacionados con las características de las actividades de los desarrolladores de software.

La actividad en esta fase no trata de describir los temas en detalle, pero si identificarlos como parte de los requisitos de conocimientos. Los temas que se

identificaron en esta fase se describen de la siguiente manera: i) *Perfil del conocimiento*, esto se refiere a describir el perfil de conocimiento de los expertos, para poder identificar qué es lo que saben y a qué nivel lo saben [21]. ii) *La información de Disponibilidad*, esto se refiere al conocimiento de las actividades que tienen relación con la actividad actual de un colega con el fin de iniciar la interacción con un experto. iii) *Ubicación del Conocimiento*, esto se refiere a la información sobre la manera en que está almacenado el conocimiento según la fuente (personas, artefactos, entidades organizacionales).

2.3 Fase 3: Identificando el Flujo del Conocimiento

Esta fase implicó la creación de un modelo de flujo de conocimiento del proceso de búsqueda de expertise en los equipos de desarrollo de software. Para este trabajo, el flujo del conocimiento está representado por un autómata finito (ver Figura 1).

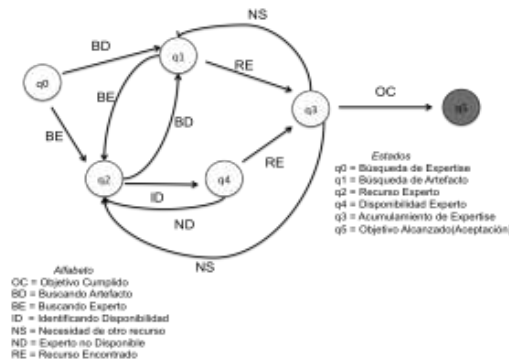


Figura 1. Representación del flujo de conocimiento mediante un autómata

El autómata (Figura 1) representa el flujo del conocimiento del proceso búsqueda de *expertise*, donde se inicia teniendo la necesidad de conocimiento para resolver alguna dificultad en una actividad (q0). Posteriormente se puede elegir entre hacer una búsqueda de artefactos (q1) o una búsqueda de un experto (q2). En el caso de elegir una búsqueda de artefactos (q1) se busca entre todos los artefactos disponibles (páginas, manuales, videos, código reutilizado), al encontrar un artefacto se verifica si este ayudó a cumplir el objetivo o aún es necesario buscar más artefactos (q1). Si no es necesario buscar más artefactos entonces el objetivo se cumplió (q5), de lo contrario se puede buscar más artefactos (q1). Un artefacto puede sugerir que se busque a un experto (q2). En el caso de elegir una búsqueda de experto (q2) se inicia buscando expertos con el grado de conocimiento para resolver la dificultad que se tienen en la actividad, una vez que se encuentra un experto se necesita comprobar su disponibilidad (q4) para iniciar una interacción con él, posteriormente se comprueba si la consulta del experto fue suficiente (q3) o se necesita consultar al experto o a otro (q2), también en el caso de estar buscando un experto puede ocurrir que un experto sugiera algún artefacto (q1), si el objetivo se cumplió el proceso termina (q5).

2.4 Fase 4: Identificando los Obstáculos en el Flujo del Conocimiento

Esta fase consiste en la identificación de los obstáculos que se presentan en el proceso de búsqueda de *expertise* los cuales fueron obtenidos a partir de las entrevistas. Esto permitió que el proceso de clasificación de los obstáculos y la búsqueda de una posible solución. Los problemas identificados se describen en la Tabla 1. La primera columna define los problemas que surgieron a partir del análisis de las fuentes y temas de conocimiento. La segunda columna describe brevemente una situación de ejemplo para ilustrar el problema.

Tabla 1. Lista de problemas y situaciones identificadas.

Problemas	Situaciones
1. Administración de los artefactos (individual o grupal)	En algunos casos se conoce al proveedor del conocimiento pero no se tiene acceso a sus artefactos (blogs, manuales, código reutilizado).
2. Administración de los Expertos	Muchas veces es difícil encontrar a la persona con el nivel adecuado de <i>expertise</i> para poder consultar alguna duda o resolver un problema en una actividad.
3. Disponibilidad de los Expertos	En algunos casos no se sabe si el <i>expertise</i> o el experto está disponible para la persona que lo está buscando.
4. Tiempo de resolución de dificultades	En algunos casos se pierde mucho tiempo en la búsqueda de <i>expertise</i> por qué no se cuenta con el conocimiento de donde se encuentra o quienes son los proveedores.

3 Implicaciones de Diseño

Con base en el conjunto de problemas de flujo de conocimiento identificados (ver Tabla 1), estos fueron transformados en características de la búsqueda de *expertise*. Por lo que en la Tabla 2 se presentan las implicaciones del diseño de un sistema que podría proporcionar un apoyo para la búsqueda de *expertise* en el desarrollo de software. La primera columna define la característica que sería deseables en el proceso de búsqueda de *expertise*. La segunda columna describe la implicación de diseño que se debe tomar en cuenta para poder dar soporte a la búsqueda de *expertise* durante las actividades del desarrollo de software.

Tabla 2. Implicaciones de diseño.

Características	Implicaciones de diseño
1. Gestión del conocimiento	<i>I1.</i> Uso de mecanismos que recolecten el conocimiento de la empresa (personas, artefactos, entidades organizacionales).
2. Búsqueda de Artefactos	<i>I2.</i> Uso de mecanismos que permitan dar acceso a los artefactos (prácticas, repositorios, tecnologías) de los miembros dentro de la empresa.
3. Búsqueda de Expertos	<i>I3.</i> Uso de mecanismos que permitan encontrar a los proveedores con el conocimiento y grado adecuado para consultar y para resolver dificultades con alguna actividad.
4. Acceso al Conocimiento	<i>I4.</i> Uso de mecanismos que permitan acceso al conocimiento (personas, artefactos y entidades organizacionales) por parte de cualquier miembro de la organización.

4 Conclusiones y Trabajos Futuros

La búsqueda de *expertise* implica la necesidad de interactuar con personas, así como también acceder a sus fuentes de conocimiento. Con la ayuda de la metodología KoFI fue posible determinar las fuentes de conocimiento en el desarrollo de software, que principalmente son las personas, puesto que son los que tienen acceso al *expertise*. Dicho *expertise* puede estar representado en blogs, videos, manuales y en los trabajadores del desarrollo de software. También fue posible identificar los elementos clave del *expertise*, que son la información relacionada a la disponibilidad de los proveedores del *expertise*, el conocimiento al que tienen acceso dichos proveedores, y la información del perfil de los proveedores. Para describir el flujo del conocimiento en el desarrollo de software se utilizó un autómata finito para ilustrar los diferentes caminos que se pueden tomar cuando se busca el *expertise*. A partir de esto se identificaron los obstáculos que se presentan durante la búsqueda del *expertise* en el desarrollo de software, los cuales fueron la administración del conocimiento, la administración de los expertos y el tiempo de respuesta para dicha búsqueda. Para abordar estos obstáculos, se proponen mecanismos que permitan gestionar el conocimiento de la organización, buscar y tener acceso a dicho conocimiento y contar con la información de la disponibilidad de los proveedores del *expertise*. Los resultados de este trabajo son la base para el desarrollo de una herramienta tecnológica para soportar la búsqueda de *expertise* con un sistema multi-agente que facilite la distribución de este tipo de información de manera transparente a los usuarios.

Referencias

1. Ammar-Khodja, S., Bernard, A.: An overview on knowledge management. *Methods and tools for effective knowledge life-cycle-management*, pp. 3-21. Springer (2008)
2. Serban, A.M., Luan, J.: Overview of knowledge management. *New Directions for Institutional Research* 2002, 5-16 (2002)
3. Prusak, L.: Where did knowledge management come from? *IBM systems journal* 40, 1002-1007 (2001)
4. Povelis, S., Fairer-Wessels, F.A.: Knowledge management: A literature overview. *South African Journal of Libraries and Information Science* 66, (2014)
5. Alavi, M., Leidner, D.E.: Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly* 107-136 (2001)
6. Wang, S., Noe, R.A., Wang, Z.-M.: Motivating Knowledge Sharing in Knowledge Management Systems A Quasi-Field Experiment. *Journal of Management* 40, 978-1009 (2014)
7. Jones, G., Sallis, E.: *Knowledge management in education: Enhancing learning & education*. Routledge (2013)
8. Petrides, L.A., Nodine, T.R.: *Knowledge Management in Education: Defining the Landscape*. (2003)

9. Abidi, S.S.R.: Knowledge management in healthcare: towards 'knowledge-driven' decision-support services. *International Journal of Medical Informatics* 63, 5-18 (2001)
10. Nicolini, D., Powell, J., Conville, P., Martinez- Solano, L.: Managing knowledge in the healthcare sector. A review. *International Journal of Management Reviews* 10, 245-263 (2008)
11. Jain, R.: Improvement in Software Development Process and Software Product through Knowledge Management. *International Journal of Computer Technology and Applications* 2, 1557-1562 (2011)
12. Rodríguez, O.M., Vizcaíno, A., Martínez, A.I., Piattini, M., Favela, J.: Using a multi-agent architecture to manage knowledge in the software maintenance process. *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 1181-1188. Springer (2004)
13. Rus, I., Lindvall, M.: Guest editors' introduction: Knowledge management in software engineering. *IEEE software* 19, 26-38 (2002)
14. Espinosa, J.A., Carmel, E.: The impact of time separation on coordination in global software teams: a conceptual foundation. *Software Process: Improvement and Practice* 8, 249-266 (2003)
15. Keil, P., Paulish, D.J., Sangwan, R.S.: Cost estimation for global software development. *Proceedings of the 2006 international workshop on Economics driven software engineering research*, pp. 7-10. ACM (2006)
16. Ericsson, K.A., Prietula, M.J., Cokely, E.T.: The making of an expert. *Harvard business review* 85, 114 (2007)
17. Rodríguez-Elias, O.M., Vizcaíno, A., Martínez-García, A.I., Favela, J., Piattini, M.: Knowledge Flow Identification. *Encyclopedia of Information Science and Technology* 2337-2342 (2009)
18. Rodríguez-Elias, O.M., Vizcaíno, A., Martínez-García, A.I., Favela, J., Piattini, M.: Studying Knowledge Flows in Software Process. In: Belini, E.A. (ed.) *Software Engineering and Development*, pp. 37-68. Nova Publishers (2009)
19. Martin, B., Hanington, B., Hanington, B.M.: *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Pub (2012)
20. Becerra-Fernandez, I., Sabherwal, R.: *Knowledge management: systems and processes*. ME Sharpe (2010)
21. Velázquez Mendoza, M.J., Rodríguez-Elias, O.M., Rose Gómez, C.E., Meneses Mendoza, S.R.: Modelo para diseño de perfiles de conocimiento: una aplicación en la industria generadora de energía eléctrica. *Research in Computing Science* 55, 125-135 (2012)