

# Comunidades de práctica en el desarrollo de software libre. CaMPI como caso de estudio.

Víctor M. Ferracutti<sup>1</sup>

<sup>1</sup>Biblioteca Central, Universidad Nacional del Sur, Argentina  
vmferra@uns.edu.ar

**Resumen.** El uso de herramientas colaborativas es especialmente importante para compartir conocimiento entre los desarrolladores y usuarios durante todo el ciclo de vida de software libre de código abierto de distribución gratuita, a través de un aprendizaje colaborativo informal. La definición y caracterización de las comunidades de práctica para luego estudiar una comunidad de práctica de software libre de código abierto denominada CaMPI permitirá ilustrar los diferentes factores que determinan la efectividad y eficiencia de dichas herramientas.

**Palabras claves.** Comunidad de práctica, desarrollo de software libre, código abierto, aprendizaje colaborativo, herramientas colaborativas.

## 1 Introducción

Las comunidades de práctica son grupos de personas que comparten una preocupación o una pasión por algo que hacen y aprenden a hacerlo mejor a partir de las interacciones regulares entre ellos [1]. Tres características fundamentales de las comunidades de práctica son:

- El dominio o interés compartido que le da identidad;
- La comunidad creada a partir de las interacciones determinadas, por ejemplo, por actividades conjuntas o discusiones;
- La práctica compartida, por ejemplo a través del intercambio de buenas prácticas o lecciones aprendidas.

Una tendencia en estas comunidades de práctica es la existencia de un núcleo de participantes cuya dedicación al interés común provee la energía suficiente para mantener y motivar a toda la comunidad. Estos participantes resultan los líderes intelectuales y sociales de la comunidad [2].

La principal diferencia de una comunidad de práctica con los equipos de trabajo, es que estos últimos los forma la gerencia de una organización y se reportan ante un jefe, tienen una membresía definida, plazos y entregables específicos. Las comunidades de práctica pueden ser voluntarias y mayormente son responsables por sí mismas, aunque suelen necesitar de instituciones que faciliten su funcionamiento.

La clave para la efectividad en estas comunidades es que funcionan en formas que mejor se ajustan a los intereses y estilos de trabajo de sus miembros en vez de adherir a modelos operativos poco flexibles.

Las comunidades de práctica se desarrollan alrededor de cosas que le interesan a la gente. Como resultado, sus prácticas reflejan la propia comprensión de sus miembros respecto de qué es lo importante. Obviamente, las restricciones o directivas externas pueden influenciar esa comprensión, pero aún así, los miembros desarrollan prácticas que son su propia respuesta a esas influencias externas. Aún cuando las acciones de una comunidad se ajustan a un mandato externo, es la comunidad –no el mandato– lo que produce la práctica [3].

Estas prácticas, llevadas adelante por los desarrolladores y usuarios de proyectos de software libre de código abierto, comparten las características mencionadas al comienzo en el sentido que:

- Existe un claro dominio compartido, que es el proyecto (representado mayormente por el código fuente);
- Los proyectos subsisten gracias una activa comunidad que comparte información y se involucra en actividades conjuntas;
- Existe una práctica compartida, al menos por los miembros de la comunidad que usan el mismo código fuente.

Tal como se señala en [4], dentro de los proyectos de software libre existe un conjunto de pasos que facilita al simple usuario transformarse en un activo participante del proyecto, tales como:

- Ganar experiencia instalando el software en su computadora o en un servidor Web;
- Contribuir a los foros de discusión;
- Contribuir a la documentación y a la promoción;
- Reportar errores y verificar las distintas versiones;
- Modificar el código para personalizar una operación o corregir un error;
- Crear un módulo para extender la funcionalidad;
- Entregar parches y módulos para revisión por pares e incorporarlos en el tronco principal del proyecto;
- Ser parte del grupo de desarrolladores con acceso al árbol CVS<sup>1</sup> con permisos de envío.

Por otra parte, y tal como define [5], una comunidad de práctica tiene un conjunto de intereses en común para realizar una tarea conjunta y no está necesariamente ubicada en un mismo lugar físico. Por lo tanto, la clave para una estrategia de diseminación efectiva del conocimiento es canalizar el conocimiento hacia la comunidad y proveer medios para el intercambio de información y la colaboración entre pares [6].

Un caso de comunidad de práctica distribuida geográficamente con necesidades de comunicación, colaboración y cooperación; es la de desarrolladores y usuarios de software libre de código abierto. Si bien existen múltiples técnicas y herramientas para el intercambio de conocimiento dentro de este tipo de comunidad virtual, los objetivos principales tienen que ver que proveer el conocimiento a los miembros que lo requieran, motivarlos para que contribuyan y facilitar la comunicación entre ellos [7].

La interacción entre usuarios y desarrolladores de software libre de código abierto es una de las principales diferencias respecto del software comercial o propietario. La realimentación provista por diferentes medios de comunicación dentro de esta comunidad de práctica informal es crítica para la mejora del producto [8].

---

<sup>1</sup> <http://savannah.nongnu.org/projects/cvs>

Esta estructura informal de organización de comunidad sin una política formal de gobierno genera una reacción negativa en los administradores de sistemas ya que no existe un único responsable del producto, tal como sucede en el software propietario.

De todos modos, diferentes casos de éxito de software libre de código abierto (Apache<sup>2</sup>, Linux<sup>3</sup>, Moodle<sup>4</sup> por citar algunos) muestran la importancia de las comunidades de práctica que los sustentan en el desarrollo y en la entrega de estos productos.

En este sentido, los proyectos de software libre de código abierto descansan en una infraestructura tecnológica que facilita la captura selectiva e integración de la información. Una infraestructura adecuada evita que el proyecto colapse en el sentido explicitado por la ley de Brooks<sup>5</sup> que establece que añadir más gente a un proyecto de software atrasado, lo atrasará aún más. El desafío entonces radica en hacer sentir a todos los miembros de una comunidad distribuida que están trabajando juntos en una misma oficina [9].

Para tal fin, la mayor parte de los proyectos de software libre de código abierto ofrecen como mínimo un conjunto de herramientas entre las que se encuentran:

- Sitio Web: para transmitir la información del proyecto al público en general;
- Listas de correo electrónico: foro de comunicación dentro del proyecto por excelencia y también el medio registral;
- Control de versiones: para administrar los cambios en el código fuente, incluyendo la posibilidad de revertir los mismos;
- Administrador de *bugs*: permite a los desarrolladores llevar registro de las actividades en las que están trabajando;
- *Chat* en tiempo real: un medio para discusiones rápidas e intercambios del tipo pregunta/respuesta.

Estas herramientas en particular y los sistemas basados en computadoras que apoyan a grupos de personas involucradas en una tarea común (*groupware*) en general, proveen una interface a un ambiente compartido para apoyar interacción miembro-

---

<sup>2</sup> <http://www.apache.org/>

<sup>3</sup> <http://www.linux.org/>

<sup>4</sup> <http://moodle.org/>

<sup>5</sup> [http://en.wikipedia.org/wiki/Brooks's\\_law](http://en.wikipedia.org/wiki/Brooks's_law)

miembro que se puede resumir en actividades de comunicación, colaboración y coordinación [10]. Las herramientas pueden clasificarse utilizando una taxonomía basada las nociones de tiempo y espacio como se observa en la Tabla 1, o bien por la funcionalidad de la aplicación (sistemas de mensajes, videoconferencia, sistemas de coordinación, etc.).

	<b>Mismo momento</b>	<b>Diferentes momentos</b>
<b>Mismo lugar</b>	Interacción cara a cara	Interacción asíncrona
<b>Diferentes lugares</b>	Interacción síncrona distribuída	Interacción asíncrona distribuída

**Tabla 1.** Clasificación según tiempo y espacio

El resto del documento se estructura de la siguiente forma: en la Sección 2 se presenta una caracterización de las comunidades de práctica describiendo el concepto de identidad. En la Sección 3 se reseña el aprendizaje colaborativo a través de las comunidades de práctica. La Sección 4 describe una comunidad de práctica de desarrollo de software libre de código abierto. En la Sección 5 se comentan las lecciones aprendidas y en la Sección 6 las conclusiones generales y el trabajo futuro.

## **2 Caracterización de las Comunidades**

Desde la perspectiva de la teoría del aprendizaje social, en donde el aprendizaje implica participación en una comunidad dejando de ser considerado como la adquisición de conocimiento por individuos para ser reconocido como un proceso de participación social en el que la naturaleza de la situación impacta significativamente, el significado y las identidades son construidos en las interacciones, mientras que la construcción de estos significados e identidades es influenciada por el contexto en el que se inscriben [11].

En las comunidades de práctica no hay separación entre el desarrollo de la identidad y el desarrollo del conocimiento, ambos interactúan recíprocamente mediante el proceso de participación periférica legítima en el contexto de una comunidad de práctica.

Según [12] la identidad es el pivote entre lo social y lo individual sosteniendo que la idea de que "la experiencia de conocer no es menos única, menos creativa y menos extraordinaria por ser una experiencia de participación". Entonces, a través de la participación es que las comunidades de práctica toman auténtica forma. El mismo autor parte de la idea de que "el aprendizaje no se puede diseñar: sólo se puede facilitar o frustrar".

La importancia de la participación, la cosificación y la identidad, como los tres elementos que convergen al igual que los lados de una pirámide en las dimensiones de una comunidad de práctica, deberán unirse al papel que deben desempeñar las organizaciones y las instituciones en la tarea de diseñarla. El compromiso de la organización, su educación y la de sus integrantes aparecen, también, como elementos básicos para conseguir una arquitectura de aprendizaje que facilite el nacimiento de las comunidades de práctica [13].

Como se mencionó en la Sección 1 las tres características cruciales que determinan una comunidad de práctica son el dominio, la comunidad y la práctica.

El dominio compartido define la identidad por medio de la membresía, que implica un compromiso con el dominio y por lo tanto una competencia que distingue a los miembros del resto de la gente. El dominio no es necesariamente algo reconocido como "experiencia" fuera de la comunidad. La comunidad valora su competencia colectiva y los miembros aprender unos de otros, a pesar de que pocas personas fuera del grupo de valor pueden reconocer su experiencia [1].

Por medio de las actividades compartidas y las discusiones, los miembros de la comunidad construyen relaciones entre ellos que les posibilitan aprender unos de otros, aún a pesar de no necesariamente compartir el trabajo diariamente. Las interacciones, independientemente del medio que se utilice, son esenciales para construir la comunidad.

Por último, y además de compartir intereses comunes (i.e., dominio) los miembros de una comunidad de práctica son esencialmente practicantes, desarrollando un repertorio de recursos compartidos (por ejemplo: experiencias, herramientas, etc.). Llevar adelante una práctica común insume tiempo y un esfuerzo consciente de cada uno de los participantes.

En el caso particular de las comunidades de desarrollo de software libre de código abierto suele existir un conjunto de pasos que facilitan al simple usuario transformarse en un activo participante del proyecto, que se han descrito en la Sección 1. La progresión de estos pasos ofrece un aprendizaje informal y la participación de los miembros se reconoce a través de sus roles asignados dentro de un proyecto.

Por otra parte, y mayormente debido a que el software es gratuito, estas comunidades tienden a presentar una cultura de donación o disposición. Los reconocimientos por contribuir a desarrollos de código abierto tienen más que ver con el reconocimiento de los pares que con retornos económicos.

### **3 Aprendizaje a través de las Comunidades de Práctica**

Tal como introduce [14] el aprendizaje visto como actividad situada tiene como característica central un proceso denominado participación periférica legítima. Esto implica que los miembros participan inevitablemente en comunidades de práctica y que el dominio del conocimiento y la destreza les exigen a los novatos acercarse a la participación plena en las prácticas socioculturales de una comunidad. “Participación periférica legítima” permite hablar de las relaciones entre novatos y veteranos y de las actividades, identidades, artefactos, y comunidades de conocimiento y práctica. Trata del proceso por el que los nuevos participantes se convierten en parte de una comunidad de práctica. El proceso de convertirse en participante pleno de una práctica sociocultural compromete los propósitos de aprender de una persona y configura el significado del aprendizaje. Este proceso social incluye, de hecho subsume, el aprendizaje de destrezas conocibles.

Cabe aclarar que el concepto de periferia legítima es una noción compleja, implicada en estructuras sociales involucradas en relaciones de poder. Como un lugar en el cual uno se mueve hacia una participación más intensa, la periferia está en una posición de poder. En cuanto lugar en el cual uno se encuentra impedido de participar plenamente, es una posición de no poder. Más allá de esto, la periferia legítima puede ser una posición en la articulación de comunidades relacionadas. En este sentido, puede ser una fuente de poder o de impotencia, al provocar o impedir la articulación y el intercambio entre las comunidades de práctica [14].

Se concibe así al aprendizaje como un proceso de participación social. Frente al implacable celo con el que actúan la mayoría de las instituciones contra las comunidades de práctica, [12] defiende que el aprendizaje basado en la participación puede ser muy beneficioso para las organizaciones.

De acuerdo a lo anterior, se está ante situaciones de aprendizaje colaborativo (CL del inglés *Collaborative Learning*), que implica que todos los participantes contribuyen a lograr una solución a los problemas propuestos como objetivo común del aprendizaje [15-16]. Y en él las interacciones sociales dentro del grupo son el mecanismo central para el logro del objetivo [16]. Un tipo concreto de aprendizaje colaborativo es aquel que se apoya en la tecnología para facilitar la colaboración entre los participantes (CSCL del inglés *Computer-Supported Collaborative Learning*). En [17] se distingue como un beneficio propio del aprendizaje basado en ordenador la posibilidad de facilitar un intercambio colaborativo y un discurso crítico reflexivo a través de la comunicación mediada por el ordenador, aunque señala como una cuestión compleja saber cómo funciona la interacción *on-line*.

#### **4 Caso de Estudio: CaMPI**

El sistema CaMPI<sup>6</sup> refiere a la integración de desarrollos de *software* consolidados de código abierto para bibliotecas con el objetivo principal de consolidar las comunidades de desarrollo y de usuarios preexistentes que permitan el mantenimiento y el rediseño de los productos de acuerdo a las nuevas herramientas de programación Web disponibles [18].

Este proyecto se basa principalmente en dos proyectos previos: Catalis<sup>7</sup> y Open MarcoPolo<sup>8</sup>. Catalis, junto a su derivado OPAC-MARC, son desarrollos de código abierto que implementan módulos de Catalogación y OPAC respectivamente, realizados por Fernando J. Gómez (INMABB, CONICET-UNS) en función a las necesidades concluidas en las Jornadas de Tecnologías de la Información<sup>9</sup> desarrolladas en la

---

<sup>6</sup> <http://campi.uns.edu.ar/>

<sup>7</sup> <http://catalis.uns.edu.ar/>

<sup>8</sup> <http://marcopolo.uner.edu.ar/>

<sup>9</sup> <http://jti.uns.edu.ar/>



Universidad Nacional del Sur (UNS) en mayo de 2000 para tal fin. Open MarcoPolo es un desarrollo de código abierto de la Universidad Nacional de Entre Ríos (UNER) que implementa los módulos de Circulación y Estadísticas de un Sistema Integrado de Gestión Bibliotecaria (SIGB). Este también es un producto con una amplia comunidad de usuarios. Tanto Catalis y OPAC-MARC como Open MarcoPolo poseen una comunidad de usuarios nacional y regional.

Previo al inicio del proyecto CaMPI en julio de 2007, se realizaron pruebas piloto de integración de ambos productos con resultados satisfactorios. Los desarrolladores de ambos productos han mostrado la voluntad de formalizar la integración, sumando recursos humanos externos para consolidar la comunidad de desarrollo. Esta consolidación tiene que ver con mejorar los tiempos de respuesta para implementar requerimientos de nuevas funcionalidades (a partir de los planteados por las comunidades de usuarios) y la carencia de una documentación adecuada, junto con características de calidad de software.

El proyecto CaMPI surgió como entidad a partir de la creación de un grupo de discusión<sup>10</sup>, la selección de un espacio en SourceForge.net<sup>11</sup> y la realización de un encuentro presencial con un grupo inicial de participantes comprometidos con el proyecto. SourceForge.net es una central de desarrollos de software que controla y gestiona varios proyectos de software libre y actúa como un repositorio de código fuente.

Este grupo inicial, con pertenencia a diferentes instituciones distribuidas geográficamente en Argentina: UNER, Comisión Nacional de Energía Atómica (CNEA), INMABB (CONICET-UNS) y UNS; definió las líneas de acción generales del proyecto a partir de la primera reunión presencial y periódicamente comunicaciones a través del *chat*. Las instituciones mencionadas conforman las instituciones patrocinantes del proyecto.

En función del avance del proyecto, se han comenzado a utilizar distintas herramientas de comunicación, colaboración y coordinación que se enumeran a continuación:

---

<sup>10</sup> [biblioinformaticos@googlegroups.com](mailto:biblioinformaticos@googlegroups.com)

<sup>11</sup> <https://sourceforge.net/projects/campi/>

- Una Wiki<sup>12</sup> del proyecto para depositar, compartir y realizar escritura colaborativa de documentación técnica y del usuario e informes de avance de las reuniones presenciales;
- Un sitio Web<sup>13</sup> basado en el sistema de gestión de contenidos Joomla! como portal del sistema CaMPI en donde se describe en general el proyecto y se establecen las condiciones de participación;
- Un grupo de discusión para los usuarios del sistema<sup>14</sup>, por medio del cual se plantean y resuelven inquietudes respecto de la operación y mantenimiento del sistema;
- Un grupo de Google denominado campi-gestion<sup>15</sup> a través del cual cada institución patrocinante y asociada (aquellas que actualmente colaboran y no han compuesto el grupo inicial) realizan la coordinación del proyecto;
- Un sistema de gestión de peticiones (tickets)<sup>16</sup> basado en Redmine<sup>17</sup> para realizar el seguimiento de los requerimientos, errores o tareas planteadas.

A los efectos de darle sustentabilidad a la comunidad (i.e. mantenerla activa), las instituciones patrocinantes promueven convenios formales entre las instituciones participantes, el uso de modelos de madurez para asegurar calidad de productos, compartir guías de buenas prácticas, el uso de espacios virtuales de comunicación y la realización de talleres presenciales periódicos que permiten mostrar los avances realizados, fomentan la interacción entre las comunidades de usuarios y desarrolladores, facilitan trabajar en diferentes aspectos de la implementación del software y definen los pasos a seguir luego de la reunión.

Toda la información generada (por ejemplo: cambios de código fuente, informes de reuniones presenciales, etc.) está públicamente accesible a través de las distintas herramientas enumeradas al comienzo de la sección.

---

<sup>12</sup> <http://campi.wiki.sourceforge.net/>

<sup>13</sup> <http://campi.uns.edu.ar/>

<sup>14</sup> <http://groups.google.com/group/usuarios-campi?hl=en>

<sup>15</sup> <http://groups.google.com/group/campi-gestion?hl=en>

<sup>16</sup> <http://ticketsbc.uns.edu.ar/projects/campi>

<sup>17</sup> <http://www.redmine.org/>

Como muestra de compromiso, todas las instituciones patrocinantes y asociadas utilizan actualmente el sistema CaMPI, intercambiando experiencias de trabajo y resolviendo inquietudes a través de las listas de correo electrónico.

En cuanto al aprendizaje colaborativo, tiene que ver tanto con el desarrollo y modificación del código fuente como con el uso del sistema. En el primer caso, los desarrolladores realizan sus actividades a través de la programación por pares<sup>18</sup> durante las reuniones presenciales y también en los momentos no presenciales (a través del repositorio en Sourceforge.net y las conversaciones de *chat*).

Las diferentes tareas de las cuales la comunidad se encarga, se han ido estableciendo con el paso del tiempo y de la misma forma cada miembro ha asumido su propio rol dentro de la comunidad. Actualmente, y en cuanto al desarrollo, se realizan las siguientes tareas:

- **Definición de requerimientos:** Realizada a través de las listas de correo de las comunidades de desarrolladores y de usuarios y del sitio <http://ticketsbc.uns.edu.ar/projects/campi>. Aplicable a las instituciones patrocinantes y asociadas. Los responsables son el Jefe de Sistemas de la Biblioteca Central de la UNS (BC-UNS) y los moderadores de las listas de correo;
- **Análisis y diseño:** Establece el conjunto de requerimientos a desarrollar en el período establecido (*sprint* de acuerdo a la metodología *Scrum*<sup>19</sup>), que concluye con la aparición de una nueva versión de CaMPI. Se realiza al menos una reunión presencial anual con la participación de las comunidades de usuarios y de desarrolladores. Los responsables de esta etapa son la comunidad de usuarios, la comunidad de desarrolladores y la institución anfitriona de la reunión presencial;
- **Codificación (programación):** A cargo de la comunidad de desarrolladores. Se incluye documentación técnica dentro del propio código, en los comentarios agregados en el repositorio de código y a través del sistema de gestión de peticiones;
- **Verificación:** Al finalizar el *sprint*, se empaqueta un instalador de CaMPI, verificado por miembros de las comunidades de usuarios y desarrolladores, llamados ve-

---

<sup>18</sup> [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_en\\_pareja](http://es.wikipedia.org/wiki/Programaci%C3%B3n_en_pareja)

<sup>19</sup> [http://en.wikipedia.org/wiki/Scrum\\_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))

rificadores (*testers*). Los verificadores no pueden haber estado involucrados en la codificación de esa versión. La verificación transcurre durante los 15 días siguientes al empaquetamiento del instalador;

- **Implantación:** Finalizada la verificación, se publica en <http://sourceforge.net/projects/campi/> el instalador empaquetado para su descarga. Aquellos usuarios que tengan instalada una versión previa de CaMPI deben actualizarla a través de la sincronización con el repositorio de código. Los responsables son el Departamento de Sistemas de la BC-UNS y la comunidad de usuarios;
- **Mantenimiento:** Se provee para instituciones patrocinantes y asociadas a través de la comunidad de desarrolladores;
- **Evaluación de la madurez de CaMPI** utilizando el *Open Source Maturity Model* (OSMM). Se realiza al menos en forma anual y con la aparición de una nueva versión y se presenta en las reuniones presenciales. El responsable de realizarla es el Jefe de Sistemas BC-UNS.

El uso progresivo de las diferentes herramientas mencionadas para facilitar la colaboración, comunicación y coordinación; así como también las mejoras sostenidas del sistema, a partir de la calificación OSMM; dan una noción de efectividad de la comunidad.

Tal como se mencionan en [18], las medidas de efectividad (o indicadores de éxito) del proyecto CaMPI son los siguientes:

- Uso efectivo de los espacios de comunicación para el entrenamiento en nuevas herramientas de desarrollo;
- Análisis y diseño de requerimientos, a través de herramientas estándares (diagramas de casos de uso, diagramas de clases, etc.);
- Uso de métricas de calidad de software de código abierto;
- Uso de un repositorio de desarrollo de requerimientos pendientes.

Por otra parte, la existencia de entregables también determina la efectividad de la comunidad. Entre los entregables definidos están:

- Documento de análisis y diseño del sistema: describe cuáles son los requerimientos funcionales del nuevo sistema y cómo se implementan;
- Sistema en funcionamiento: incluye el software del servidor y un instalador para versiones de Microsoft Windows y Linux con una configuración básica;
- Manual de instalación del sistema: describe los pasos que deberán seguirse para instalar exitosamente en un servidor el sistema;
- Manual de uso del sistema: describe los pasos que deberán seguirse para las distintas tareas provistas por el sistema;
- Cursos de capacitación: incluyen la capacitación del personal de las áreas de Procesos Técnicos y Atención al Público en el uso del sistema, la capacitación a los usuarios del nuevo sistema y la capacitación al personal de desarrollo informático;
- Conversión de datos: incluye la estandarización, normalización y migración de los datos de la primera versión operativa de CaMPI;
- Ejercicios de prueba: incluyen ejemplos con los datos básicos provistos por el instalador del sistema;
- Repositorio de software: que contiene el código fuente del desarrollo y un Wiki con documentación de programación (estructura de bases de datos, diagramas de casos de uso y descripciones respectivas) e informes de avance;
- Listas de discusión: para usuarios y para desarrolladores;
- Documento de prueba: con casos de prueba y política de testeo.

## **5 Lecciones Aprendidas**

Las comunidades de práctica se consideran una técnica de adquisición y gestión del conocimiento. Las herramientas para efectuar la gestión del conocimiento utilizadas por estas comunidades son representadas en una variedad de implementaciones tales como repositorios de documentos, bases de datos de expertos, listas de discusión, sistemas de recuperación sensible al contexto e incorporan tecnologías de filtrado colaborativo.

Estas comunidades colaboran con los objetivos de la gestión del conocimiento en tanto y en cuanto permiten explotar el conocimiento que poseen las instituciones,

renovar el conocimiento de las personas y organizaciones y transformar el conocimiento tácito en conocimiento explícito.

Algunos de los factores que han impulsado la gestión del conocimiento tienen que ver con la necesidad de difundir y compartir las experiencias, incorporar técnicas existentes y reducir tiempos de respuesta. Si bien estos factores habitualmente son pensados dentro de una organización, en el caso particular del desarrollo de software libre de código abierto las distintas cuestiones de gestión del conocimiento trascienden las fronteras de una empresa e involucran a un conjunto de voluntarios distribuidos geográficamente. El progreso de estas comunidades de software libre se ha beneficiado de las inversiones masivas en tecnologías de la información y comunicaciones, utilizando habitualmente servicios gratuitos tales como servidores de correo electrónico, herramientas de *chat* o repositorios de código fuente.

En este sentido, contar con una comunidad de desarrolladores más allá de la propia organización que requiere del software, posibilita disminuir los costos asociados con la reestructuración y rotación de personal; y contar con una comunidad de usuarios extendida permite realizar verificaciones (*testing*) ante una mayor cantidad de usuarios, actividad que eventualmente colabora con la difusión del software.

Una de las maneras principales de compartir experiencias en las comunidades de práctica es a través del aprendizaje. Se privilegia en estos contextos el aprendizaje colaborativo por sobre el aprendizaje individual tradicional. Una de las formas de facilitar este aprendizaje colaborativo es con el uso de apoyo telemático que sustente la comunicación, colaboración y coordinación de las actividades de aprendizaje. Si bien la evaluación de usabilidad aporta indicadores de efectividad de las aplicaciones utilizadas, esta es una tarea especialmente compleja debido al diseño pedagógico, las características del contexto y las diferentes interacciones entre los usuarios.

Además de evaluar los sistemas de gestión del conocimiento desde el punto de vista de la usabilidad, tal como lo señala una de las características del software presentadas en la norma ISO 9126<sup>20</sup>; otros enfoques de la evaluación tienen que ver con el impacto en la organización, la eficiencia de las tareas usando el software y la interacción de los usuarios. En este último caso suelen utilizarse métodos mixtos (cuantitati-

---

<sup>20</sup> [http://en.wikipedia.org/wiki/ISO/IEC\\_9126](http://en.wikipedia.org/wiki/ISO/IEC_9126)

vos y cualitativos) de evaluación, y no es sencillo obtener conclusiones generales. Una posibilidad es explorar la utilización de herramientas del análisis de redes sociales para el análisis de interacciones.

## **6 Conclusiones**

En el presente trabajo se ha presentado la noción de comunidad de práctica y sus características principales, y en particular para comunidades de desarrollo de software libre de código abierto, ejemplificándose los conceptos con un caso de estudio.

Si bien las comunidades de práctica están caracterizadas a partir del dominio, la comunidad y la práctica; el autor ha encontrado dificultades para identificar como tal a la comunidad de práctica del caso de estudio. Esto se debe a que esta comunidad presenta también características propias de los equipos de trabajo y de las comunidades de interés. Una mayor profundización en estas cuestiones es parte del trabajo futuro.

El hecho de que las comunidades de software libre de código abierto tienen voluntarios entre sus miembros y las mismas excedan los límites de una organización, implica que el establecimiento de una metodología para la mejora de procesos usando gestión del conocimiento –si bien es considerado necesario– presenta múltiples desafíos. Estos desafíos están relacionados con formas de trabajo heterogéneas y con culturas diferentes. Queda pendiente revisar bibliografía sobre comunidades de práctica más allá de una organización y del impacto de metodologías de mejoras de procesos en estos casos.

Una de las expectativas en este trabajo ha sido la de establecer las herramientas apropiadas para facilitar el funcionamiento de las comunidades de software libre de código abierto, siendo la efectividad y eficiencia los indicadores clásicos en este sentido. Sobre el primero de ellos se ha encontrado bibliografía que fundamenta el uso de cierto tipo de software, aunque la evaluación de herramientas colaborativas para este tipo de comunidades todavía requiere desarrollo y más aún para establecer la eficiencia de las herramientas. El análisis de interacciones puede proveer indicadores para tal fin.

## 7 Referencias

1. Communities of Practice. A Brief Introduction. <http://www.ewenger.com/theory/>
2. Burk M.: Communities of Practice. En: Public Roads, Vol. 63, Num. 6 (2000)
3. Wenger E.: Communities of Practice: Learning as a Social System. En: Systems Thinker, Vol. 9, Num 5 (1998)
4. Berry M.: Open Source Projects as Communities of practice. (2009)
5. Hildreth P, Kimble C, Wright P.: Communities of Practice in the Distributed International Environment. En: Journal of knowledge Management, Vol. 4, Num. 1, pp. 27-37 (2000)
6. Wenger E.: Communities of Practice: The Key to Knowledge Strategy. En: Lesser E, Fontaine M, Slusher J, editors. Knowledge and Communities. Boston, Butterworth-Heinemann (2000)
7. Pickles, T.: Practice Guide: Techniques for Engaging with Members (2003)
8. Golden B.: Succeeding with Open Source Addison-Wesley Professional (2004)
9. Fogel K.: Producing Open Source Software. How to Run a Successful Free Software Project. O'Reilly Media (2005)
10. Ellis CA, Gibbs SJ, Rein GL.: Groupware: Some Issues and Experiences. En: Association for Computing Machinery. Communications of the ACM, Vol. 34, Num. 1 (1991)
11. Garrido A. El Aprendizaje como Identidad de Participación en la Práctica de una Comunidad Virtual. Universitat Oberta de Catalunya (2003)
12. Wenger E. Comunidades de Practicas: Aprendizaje, Significado e Identidad. Barcelona, Paidós Ibérica (2001)
13. Sanz S.: Reseña del libro Comunidades de Práctica: Aprendizaje, Significado e Identidad de Etienne Wenger (2003)
14. Wenger E, Lave J.: Situated learning: Legitimate Peripheral Participation. New York, Cambridge University Press (1991)
15. Dillenbourg P.: What do you mean by 'collaborative learning'? En: Dillenbourg P, editor. Collaborative-learning: Cognitive and Computational Approaches. Oxford, Elsevier, pp. 1-19 (1999)
16. Dillenbourg P.: Overscripting CSCL: The Risks of Blending Collaborative Learning with Instructional Design. In: Kirschner PA, editor. Three Worlds of CSCL. Can we Support CSCL? Open Universiteit Nederland, pp. 61-91 (2002)
17. Nichols M. E-learning in Context. Auckland, Laidlaw College (2008)



18. Ferracutti V.: Sistema CaMPI: Cooperación en el Desarrollo Open Source. Integración MarcoPolo – Catalis. En 38° JAIIO, Sociedad Argentina de Informática, Mar del Plata (2009)