

CaMPI. Comunidad de práctica como medio para el trabajo cooperativo en la integración de sistemas automatizados para bibliotecas

Victor Ferracutti¹; Marcelo Casquete²; Ricardo Piriz³

Resumen: El sistema CaMPI refiere a una comunidad de práctica, geográficamente distribuida, cuyo dominio es la integración de desarrollos de software consolidados de código abierto para bibliotecas en función de las necesidades explícitas de los usuarios y las nuevas herramientas de programación Web disponibles. El proceso de diseño y desarrollo de CaMPI, certificado según la norma IRAM-ISO 9001:2008, incluye el uso de herramientas colaborativas 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 caracterización de la comunidad de práctica de CaMPI permitirá ilustrar los diferentes factores que determinan la efectividad y eficiencia de dichas herramientas.

¹ Biblioteca Central, Universidad Nacional del Sur, Argentina. La correspondencia debe dirigirse a vmferra@uns.edu.ar

² Colegio de Abogados y Procuradores de Bahía Blanca, Argentina.

³ Biblioteca Central, Universidad Nacional del Sur, Argentina.

I. Introducción

CaMPI es una propuesta de sistema de gestión de bibliotecas bajo el concepto de software libre de código abierto de distribución gratuita, inicialmente integrada por desarrolladores argentinos con experiencia previa en el área, y actualmente sustentada en una comunidad de práctica.

Tal como se resume en (C. Boeris, Castorina, Ferracutti, & Franco, 2011; V. Ferracutti, 2009; V. Ferracutti, 2011), 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 (Wenger, 2006). Tres características fundamentales de una comunidad 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 (Burk, 2000).

En el caso del desarrollo de software libre de código abierto, las prácticas llevadas adelante por la comunidad de desarrolladores y de usuarios comparten las características mencionadas al comienzo en el sentido de 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 (Berry, 2009), 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 CVS⁴ con permisos de envío.

⁴ <http://savannah.nongnu.org/projects/cvs>

Sin bien las comunidades de práctica pueden ser voluntarias y mayormente responsables por sí mismas, suelen necesitar de instituciones que faciliten su funcionamiento (Wenger, 1998). En el desarrollo de software libre, cuyas comunidades de desarrolladores y usuarios suelen estar distribuidas geográficamente, las instituciones ayudan a cubrir las necesidades de comunicación, colaboración y cooperación. Dentro de las múltiples técnicas y herramientas para el intercambio de conocimiento dentro de este tipo de comunidad virtual, los objetivos principales tienen que ver con la provisión de conocimiento a los miembros que lo requieran, la motivación para que éstos contribuyan y la acción de facilitar la comunicación entre ellos (Pickles, 2003).

Por otra parte, y tal como define (Hildreth, Kimble, & Wright, 2000), 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 ese conocimiento hacia la comunidad y proveer medios para el intercambio de información y la colaboración entre pares (Wenger, 2000).

Los recursos más importantes dentro de las comunidades de desarrolladores y usuarios son la experiencia y las habilidades y los conocimientos especializados, pero la mera presencia de expertos en un equipo es insuficiente para producir trabajos de alta calidad. Este conocimiento táctico (experiencia, habilidades) debe ser administrado y coordinado con el fin de aprovechar su potencial. Es decir, las comunidades deben ser capaces de gestionar efectivamente sus interdependencias de habilidades y conocimientos a través de la coordinación de la experiencia. Esto es inherentemente un proceso social (Faraj & Sproull, 2000).

Para coordinar y controlar el adecuado funcionamiento de la comunidad, que incluye entre otras cosas el desarrollo de software, se ha documentado el proceso de diseño y desarrollo de CaMPI en el marco de un Sistema de Gestión de Calidad certificado según la norma IRAM-ISO 9001:2008 (V. M. Ferracutti, 2013a; V. M. Ferracutti, 2013b). Este marco de mejora continua para satisfacción de los usuarios, brinda un conjunto de elementos de planificación y medición que facilitan la gestión de la comunidad de práctica (Chrissis, Konrad, & Shrum, 2003; Golden, 2004).

El resto del documento se estructura de la siguiente forma: en la sección II se caracteriza al sistema CaMPI, en la sección III se presenta el proceso de diseño y desarrollo de CaMPI, en la sección IV se enuncian las herramientas computacionales colaborativas utilizadas, en la sección V se muestran los criterios de efectividad y eficiencia en el uso de las mismas y en la sección VI se comentan las lecciones aprendidas.

II. Caracterización de CaMPI

El sistema CaMPI⁵ 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 (V. Ferracutti, 2009; V. Ferracutti, 2011).

⁵ <http://campi.uns.edu.ar/>

Este proyecto se basa principalmente en dos proyectos previos: Catalis⁶ y Open MarcoPolo⁷. Catalis, junto a su derivado Oparcmarc, son desarrollos de código abierto que implementan módulos de Catalogación y OPAC respectivamente en función a las necesidades concluidas en las Jornadas de Tecnologías de la Información⁸ desarrolladas en la Universidad Nacional del Sur (UNS) en mayo de 2000 para tal fin. Open MarcoPolo es un desarrollo de código abierto 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 Oparcmarc 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 discusión⁹, la selección de un espacio en SourceForge.net¹⁰ 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.

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 de software colaborativo (*groupware* en inglés).

⁶ <http://catalis.uns.edu.ar/>

⁷ <http://marcopolo.uner.edu.ar/>

⁸ <http://jti.uns.edu.ar/>

⁹ biblioinformaticos@googlegroups.com

¹⁰ <https://sourceforge.net/projects/campi/>

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 pares¹¹ durante las reuniones presenciales y también en los momentos no presenciales (a través del repositorio en Sourceforge.net y las conversaciones de chat).

Otro ejemplo de este tipo de aprendizaje lo constituye el propio software. El módulo de catalogación Catalis ha sido utilizado en diferentes ámbitos educativos y en diferentes cursos como una herramienta para la enseñanza de la catalogación, y específicamente del formato MARC 21 (C. E. Boeris, 2009).

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.

III. Proceso de diseño y desarrollo de CaMPI

Actualmente, el proceso de diseño y desarrollo de CaMPI se encuentra certificado bajo la norma IRAM-ISO 9001:2008 en el marco del Sistema de Gestión de Calidad de la BC UNS.

Tal como puede observarse en la Ilustración 1, este proceso incluye las siguientes siete (7) etapas:

- Definición de requerimientos;
- Selección de requerimientos a implementar;
- Diseño y Codificación (programación);
- Verificación y validación;
- Distribución;
- Evaluación de la madurez de CaMPI utilizando el *Open Source Maturity Model*¹²;
- Capacitación.

Cada etapa está bajo la responsabilidad de miembros de la comunidad de práctica, como por ejemplo:

- La comunidad de desarrolladores tiene a su cargo la etapa de Diseño y Codificación, utilizando el software de colaboración para compartir documentos de diseño y el código escrito;
- Los *testers* (verificadores) realizan la Verificación de la versión preliminar del instalador de la nueva versión, entregando eventualmente un reporte de errores a ser corregidos inmediatamente;
- La comunidad de usuarios participa de la Definición de requerimientos y recibe la Capacitación una vez distribuida la nueva versión del software.

¹¹ http://es.wikipedia.org/wiki/Programaci%C3%B3n_en_pareja

¹² https://en.wikipedia.org/wiki/OpenSource_Maturity_Model

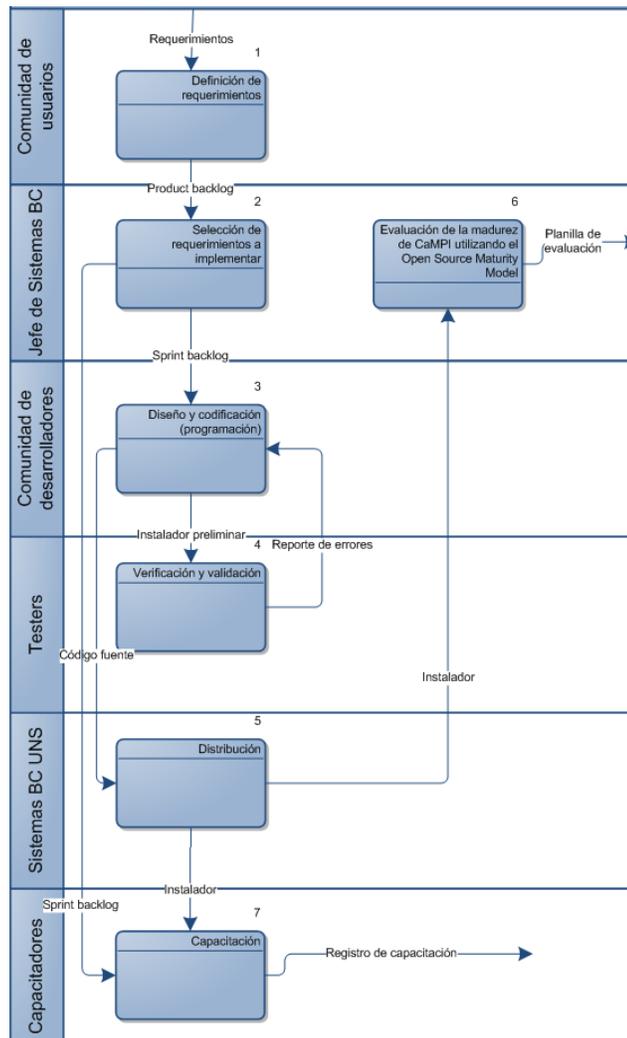


Ilustración 1. Proceso de Diseño y Desarrollo de CaMPI

Cabe aclarar que cada una de las etapas tiene indicadores asociados, cuyo cálculo es posible gracias a las herramientas colaborativas utilizadas. Entre los indicadores se incluyen, entre otros, los siguientes:

- Cantidad de requerimientos codificados
- Número de errores
- Calificación de la nueva versión
- Cantidad de usuarios capacitados

IV. Herramientas colaborativas

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

- Un sitio Web¹³ basado en el sistema de gestión de contenidos Joomla!¹⁴ 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 sistema¹⁵, por medio del cual se plantean y resuelven inquietudes respecto de la operación y mantenimiento del sistema;
- Un grupo de discusión para los desarrolladores¹⁶ por medio del cual se plantean y resuelven inquietudes respecto de las diferentes etapas del proceso de diseño y desarrollo del sistema;
- Un grupo de Google denominado campi-gestion¹⁷ 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)¹⁸ basado en Redmine¹⁹ para realizar el seguimiento de los requerimientos, errores o tareas planteadas;
- Una Wiki²⁰ del proyecto para depositar, compartir y realizar escritura colaborativa de documentación técnica y del usuario e informes de avance de las reuniones presenciales.

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.

V. Criterios de efectividad y eficiencia

Una de las medidas relevantes del sistema CaMPI es la evaluación mediante el modelo de madurez para software *open source* seleccionado, que define las siguientes seis características importantes que debe poseer un software *open source* de calidad (Golden, 2005):

- Software
- Soporte técnico
- Documentación
- Entrenamiento
- Integración
- Servicios profesionales

Si bien en el esquema de calificación se contemplan por separado cada uno de los elementos²¹, en la Tabla 1 se muestra la evolución de las calificaciones totales en los últimos años.

Año	Calificación
2008	53

¹³ <http://campi.uns.edu.ar/>

¹⁴ <http://www.joomlaspanish.org/>

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

¹⁶ <https://groups.google.com/forum/#!forum/biblioinformaticos>

¹⁷ <http://groups.google.com/group/campi-gestion?hl=en>

¹⁸ <http://ticketsbc.uns.edu.ar/projects/campi>

¹⁹ <http://www.redmine.org/>

²⁰ <http://campi.uns.edu.ar/wiki/doku.php>

²¹ <http://campi.uns.edu.ar/evaluacion>

2009	57
2010	62
2011	65
2012	68
2013	66
2014	70

Tabla 1. Calificación de CaMPI

Por otra parte, y tal como se mencionan en (V. Ferracutti, 2009; V. Ferracutti, 2011), las demás 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.

De las herramientas de apoyo a la colaboración, comunicación y coordinación utilizadas, es posible medir su efectividad de acuerdo a los miembros suscriptos a cada una de las listas, los mensajes enviados a través de las mismas, las páginas editadas en el sitio Web y en la Wiki, así como también la cantidad de revisiones de código en el repositorio y las peticiones realizadas a través del Sistema de Gestión de Peticiones.

El repositorio de código en Sourceforge.net posee 15 miembros y 526 revisiones de código. Una revisión o versión de código fuente, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. Estas revisiones indican corrección de errores y mejoras realizadas desde 2007 hasta la fecha. Como resultado la versión actual que se distribuye de CaMPI es la 1.1.0.

Las correcciones y modificaciones que figuran en el repositorio y dan lugar a las nuevas versiones del producto, son consecuencia de la resolución de peticiones o incidentes reportados en el Sistema de Gestión de Peticiones (SGP). Estas peticiones son creadas por alguno de los 36 miembros del proyecto CaMPI registrados en el SGP, que actualmente contiene 115 peticiones categorizadas de la siguiente manera:

- Tareas: 64 abiertas / 184
- Errores: 24 abiertas / 56
- Requerimientos: 23 abiertas / 27

La lista de usuarios de CaMPI contiene 61 miembros que han creado más de 200 mensajes categorizados en 105 temas. Esto implica un promedio 2,96 mensajes por tema. Esta lista se complementa con la lista de usuarios de Catalis (módulo de catalogación de CaMPI) que posee 283 miembros, quienes crearon más de 2300 mensajes categorizados en 388 temas.

En cuanto a la lista de desarrolladores de CaMPI, posee 69 miembros quienes redactaron más de 2000 mensajes distribuidos en 421 temas.

Por último, la lista de gestión de CaMPI tiene 8 miembros y más de 100 mensajes distribuidos en 59 temas.

Todos estos valores dan una pauta de la participación y la actividad en cada una de las comunidades (usuarios y desarrolladores) de CaMPI.

VI. 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, listas de discusión y sistemas de gestión de peticiones.

Algunos de los factores que han impulsado el aprendizaje colaborativo en la comunidad de práctica del sistema CaMPI 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.

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.

El uso de herramientas colaborativas (wikis, repositorios, etc.) produce un efecto de retroalimentación a la comunidad de usuarios y desarrolladores que redundará en beneficios tanto para el desarrollo como para el uso del producto.

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. Un proceso bien definido, en el marco de un Sistema de Gestión de Calidad colabora en resolver dichos desafíos.

Resta aun determinar parámetros que permitan establecer cuáles de todas las herramientas colaborativas utilizadas son las más eficientes, en función de los objetivos establecidos, tal de concentrar mayores esfuerzos en la promoción y uso de las mismas.

Bibliografía

Berry, M. (2009, 2009/9/27). Open source projects as communities of practice.

Boeris, C., Castorina, J., Ferracutti, V., & Franco, M. (2011). Sistema CaMPI. comunidad de práctica en el desarrollo de software libre para bibliotecas. *Work-Shop Software Libre O Gratuito De Gestión Bibliotecaria*, Buenos Aires.

Boeris, C. E. (2009). El uso de catalis como herramienta didáctica en la enseñanza de los procesos técnicos.

Burk, M. (2000, May/June 2000). Communities of practice. *Public Roads*, 63 Retrieved from <http://www.fhwa.dot.gov/publications/publicroads/00mayjun/commprac.cfm>

Chrissis, M. B., Konrad, M., & Shrum, S. (2003). *CMMI guidelines for process integration and product improvement* Addison-Wesley Longman Publishing Co., Inc.

Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management Science*, , 1568.

Ferracutti, V. (2009). Sistema CaMPI: Cooperación en el desarrollo open source (integración MarcoPolo – catalis). Mar del Plata.

Ferracutti, V. (2011). Comunidades de práctica en el desarrollo de software libre. CaMPI como caso de estudio. Córdoba.

Ferracutti, V. M. (2013a). *Manual de calidad de la biblioteca central "Profesor nicolás matijevic" de la universidad nacional del sur* (3 ed. ed.). Bahía Blanca:

Ferracutti, V. M. (2013b). *Política de la calidad de la biblioteca central "Profesor nicolás matijevic" de la universidad nacional del sur* (3 ed. ed.). Bahía Blanca:

Golden, B. (2005). *Succeeding with open source* Addison-Wesley Professional.

Golden, B. (2004). *Succeeding with open source (addison-wesley information technology series)* Addison-Wesley Professional.

Hildreth, P., Kimble, C., & Wright, P. (2000). Communities of practice in the distributed international environment. *Journal of Knowledge Management*, 4(1), 27-37. Retrieved from <http://arxiv.org/ftp/cs/papers/0101/0101012.pdf>

Pickles, T. (2003). Practice guide: Techniques for engaging with members.

Wenger, E. (1998, June 1998). Communities of practice: Learning as a social system. *Systems Thinker*, 9 Retrieved from <http://www.co-i-l.com/coil/knowledge-garden/cop/lss.shtml>

Wenger, E. (2000). Communities of practice: The key to knowledge strategy. In E. Lesser, M. Fontaine & J. Slusher (Eds.), *Knowledge and communities* (). Boston: Butterworth-Heinemann.

Wenger, E. (2006). Communities of practice. a brief introduction. Retrieved from <http://www.ewenger.com/theory/>