

Israel Herraiz Tabernero,
Juan José Amor Iglesias,
Álvaro del Castillo San Félix
*Grupo de Sistemas y Comunicaciones, Uni-
versidad Rey Juan Carlos, Madrid*

<{herraiz, jiamor, aacs}@gsyc.es>

Software libre para la gestión de proyectos de investigación



Herraiz, Amor y del Castillo, 2007. Este artículo se distribuye bajo la licencia "Reconocimiento-Compartir bajo la misma licencia 2.5 Genérica" de Creative Commons, disponible en <http://creativecommons.org/licenses/by-sa/2.5/deed.es_CO>.

1. Introducción

En el ámbito del Sexto Programa Marco de la Comisión Europea (6FP), el software libre ha empezado a causar interés, con el fin de ganar conocimientos y mejorar el desarrollo del software. Muchas de las buenas prácticas del software libre pueden adaptarse para la gestión de entornos complejos, aunque no tengan relación con este tipo de software. Uno de esos entornos sería el de los proyectos de investigación.

En un proyecto de investigación, personas de diferentes países trabajan de forma coordinada para conseguir llevar a cabo los objetivos del proyecto. Estas personas necesitan trabajar muchas veces en los mismos documentos o, si se trata de desarrollo de software, en los mismos ficheros de código fuente; estando estas personas geográficamente dispersas, siendo necesario que trabajen de forma coordinada con todos los demás socios, con el fin de hacer un desarrollo eficiente.

Sin embargo, los proyectos de investigación suelen llevarse de forma bastante opaca. Cada socio no está informado del trabajo que realizan los demás, y el público general accede solo a publicaciones seleccionadas, llamadas *entregables públicos* (del inglés *deliverable*); existiendo también una colección de estos entregables que no se ponen a disposición del público.

Esto es un problema importante. En primer lugar, los proyectos de investigación, al menos aquellos del Sexto Programa Marco, son financiados con fondos públicos. Los resultados, por tanto, deberían ser completamente públicos (y no solamente una selección de éstos).

Por otro lado, al menos en los proyectos del Sexto Programa Marco relacionados con el software libre, muchos comparten objetivos, y necesitan a veces las mismas fuentes de información. Estos proyectos podrían beneficiarse de otros similares si pudieran acceder a documentación interna y a otras informaciones generadas. Pensando en una analogía con el mundo del software libre, si un programador sabe que puede reutilizar un

Resumen: tradicionalmente, los proyectos de investigación se gestionan de forma bastante opaca, siendo visibles desde fuera únicamente los documentos de resultados de carácter público. Tampoco se hace pública la información sobre la evolución del proyecto, excepto para su difusión en entornos limitados. Incluso, cuando un proyecto es gestionado entre varias entidades diferentes, los detalles sobre el desarrollo de las partes asignadas a cada entidad son privados y no conocidos por los demás socios participantes. En este sentido, los proyectos de investigación resultan similares a los modelos tradicionales cerrados de desarrollo. Los proyectos de investigación IST (Information Society Technologies - Tecnologías para la Sociedad de la Información) comparten ciertas características con los proyectos de software libre, como un modelo de desarrollo distribuido global y la posibilidad de teletrabajo. En este artículo presentamos una propuesta para gestionar proyectos de investigación de este tipo, adoptando métodos utilizados por la comunidad de software libre, utilizando para ello las herramientas libres usuales de esa comunidad. Nuestra metodología hace posible los flujos de comunicación entre los diferentes socios del proyecto, incluso si se encuentran geográficamente dispersos, y además permiten hacer públicos documentos seleccionados. Creemos ciertamente que esta nueva forma de gestionar proyectos tiene importantes ventajas sobre métodos tradicionales, y puede mejorar los procesos de gestión de los proyectos de investigación.

Palabras clave: gestión de la investigación, investigación, Programa Marco, software libre.

Autores

Israel Herraiz Tabernero realiza estudios de doctorado en la Universidad Rey Juan Carlos. Su investigación está relacionada con la evolución de proyectos de software libre. En particular, está empleando análisis de series temporales y otras técnicas estadísticas para caracterizar y predecir la evolución de proyectos de software libre. Ha participado en diferentes proyectos financiados por el Programa Marco de la Comisión Europea (QUALOSS, FLOSSMetrics, Qualipso, CALIBRE). Además, ha colaborado también en otros proyectos financiados por empresas como Vodafone o Telefónica. Ha participado en la redacción de manuales y documentos sobre cómo gestionar y poner en marcha proyectos de software libre. Por ejemplo, junto con Juan José Amor y Gregorio Robles escribió un manual para el Máster en Software Libre de la Universitat Oberta de Catalunya. En este momento disfruta de un contrato de Formación de Personal Investigador, concedido en 2005 por la Comunidad de Madrid para desarrollar la tesis doctoral en el estudio de proyectos de software libre. Ha sido revisor para, entre otras conferencias, la IEEE Africon 2007, y para la revista IEEE Transactions on Software Engineering. En estos momentos está coordinando el programa del Máster en Software Libre, título propio de la Universidad Rey Juan Carlos que se imparte en colaboración con la empresa Igalia y Caixa Nova, e imparte clases en diversas titulaciones en la citada Universidad.

Juan José Amor Iglesias es Licenciado en Informática por la Universidad Politécnica de Madrid y actualmente es jefe de proyectos en la Universidad Rey Juan Carlos, trabajo que compagina con sus estudios de doctorado en la misma universidad. Sus intereses investigadores abarcan diversos temas en la Ingeniería del Software Libre, centrándose en la estimación de actividad y de esfuerzo de los desarrolladores de proyectos de software libre. Su relación con la Comunidad de Software Libre ha sido intensa desde hace años: desde 1995 ha colaborado en diversas organizaciones relacionadas con el Software Libre, siendo cofundador de LuCAS, el portal de documentación libre más conocido e Hispalinux, la asociación de usuarios españoles de software libre. Colabora también con medios como Barrapunto.com y Linux+.

Álvaro del Castillo San Félix participa en el mundo del software libre desde hace ya más de 12 años donde tuvo sus primeras experiencias con el núcleo Linux. Desde entonces su vida laboral ha estado muy ligada a este mundo en el que ha participado en proyectos como la fundación de Barrapunto.com, GNOME Hispano o Mono Hispano. Es desarrollador de GNOME habiendo participado principalmente en el desarrollo de Planner y su integración con Evolution, y dentro de las empresas en las que ha trabajado, ha estado coordinando proyectos como la primera versión de LinEx o Compatiblelinux.org. Ha trabajado como director de Desarrollo y Arquitectura en la compañía LambdaUX Software Services que tenía entre sus objetivos llevar el software libre a los usuarios finales. Durante cuatro años fue profesor asociado de la Universidad Rey Juan Carlos. Actualmente, es coordinador de proyectos europeos dentro del grupo de investigación GSyc/LibreSoft de la Universidad Rey Juan Carlos.

trozo de código de otro proyecto, simplemente acudiré a este, copiaré ese trozo y lo adaptaré a sus propias necesidades.

Por todas estas razones, proponemos una metodología para adoptar las prácticas realizadas por la comunidad de desarrollo de software libre, en la gestión de proyectos de investigación. Nuestra metodología está pensada para ser adoptada por cada uno de los socios de estos proyectos.

El resto del artículo se organiza como sigue. En la **sección 2** se describen las características de un proyecto de desarrollo típico. En la **sección 3** se describen las necesidades de un proyecto de investigación y una propuesta sobre las herramientas que sirven para esas necesidades. En la **sección 4** se explica cómo organizar el trabajo y el entorno para hacer el trabajo, basado en la experiencia de nuestro grupo. Finalmente, en la **sección 5** se incluyen algunas conclusiones.

2. Estructura de un proyecto de investigación

En esta sección describiremos la estructura de un proyecto de investigación típico. Tomaremos como ejemplos algunos proyectos del Sexto Programa Marco en los que hemos participado.

Los proyectos de investigación son propuestos y desarrollados normalmente por una serie de socios localizados en diferentes países. Esto origina el primer problema: el idioma. Normalmente, se elige el inglés como idioma para la comunicación entre socios, y para escribir los documentos resultantes del proyecto (sean internos o públicos).

El trabajo se divide en *paquetes de trabajo*. Cada socio puede liderar uno o más paquetes de trabajo, y todos los socios participarán en al menos un paquete de trabajo. Dentro de estos paquetes de trabajo encontraremos tanto *hitos* como *entregables*. Los hitos son fechas clave, en las que cierto trabajo debe ser entregado. Los entregables son documentos (aunque también puede ser un software, una base de datos, etc) que son parte del producto final del proyecto. Algunos entregables son públicos, mientras que otros son internos de los socios y otros se destinan a ser entregados al *patrocinador* del proyecto (que, en el caso del Sexto Programa Marco, es la Comisión Europea).

El trabajo requerido para realizar los entregables debe ser ejecutado habitualmente por varios socios, de forma coordinada. Normalmente, uno de los socios actúa como coordinador, siendo el que administra la parte económica, y asegurándose de que todos los trabajos que deben realizar los socios son finalizados de acuerdo con el plan de trabajo y en su debida fecha.

La coordinación es clave en los proyectos de investigación: cada socio tiene que trabajar coordinadamente con los demás, y es muy importante que todos los socios estén al tanto del trabajo del resto. Por supuesto, cada uno de los socios también es responsable de su propio trabajo y de las fechas de entrega de éste.

3. Necesidades de un proyecto de investigación

Se necesita el apoyo de ciertas herramientas para realizar el trabajo descrito anteriormente. En primer lugar, expondremos ciertos conceptos sobre lo que necesita un proyecto de investigación, y luego daremos una lista de programas libres que pueden utilizarse para satisfacer estas necesidades.

■ Sitio web

En primer lugar, se necesita un sitio web para cubrir la necesidad de *diseñar* los resultados públicos del proyecto. Suele usarse un CMS (Sistema de Gestión de Contenidos), para facilitar la participación de todos los socios en la gestión del sitio web para publicar documentos.

Tomando la definición de Wikipedia sobre CMS [1]: *Un Sistema de gestión de contenidos (Content Management System, en inglés, abreviado CMS) permite la creación y administración de contenidos principalmente en páginas web. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo el público.*

El sitio web debe además distinguir entre documentos públicos y privados, haciendo que estos últimos estén accesibles únicamente a determinados grupos de usuarios (normalmente, los socios del proyecto).

■ Listas de correo electrónico

En segundo lugar, con el fin de facilitar la comunicación entre socios, se necesitan las listas de correo electrónico. A veces es buena idea tener varias listas para cada proyecto. Una de éstas debería incluir a todas las personas involucradas en el proyecto. Otra debe incluir solamente a los máximos responsables del mismo. En nuestra opinión, algunas decisiones estratégicas sobre el proyecto deben ser discutidas únicamente por los responsables y no por todo el grupo de investigadores involucrados.

Si el grupo de personas que trabajan conjuntamente es mayor de 4 ó 5, está plenamente justificado usar una lista de correo. Pero además, las listas proporcionan otras ventajas como el archivado de mensajes, que puede ayudar a nuevos miembros del proyecto a adquirir el contexto necesario. Normalmente, habrá una lista de correo para el grupo de responsables máximos y otra para todos los involucrados. Si el grupo de investigación es muy pequeño, con una sola lista podría ser suficiente.

■ Sistema de control de revisiones

Otro requisito es tener un repositorio de documentos de trabajo, programas o ficheros de cualquier clase en general, con capacidad para controlar revisiones. Esto facilita la recuperación de versiones antiguas de los documentos y trabajar varias personas a la vez en el mismo documento de forma coordinada. Es también un lugar donde cualquiera puede encontrar cualquier documento o fichero del proyecto. El repositorio no está pensado para la publicación de entregables, sino más bien para realizar el trabajo de desarrollo cooperativo. Las capacidades de control de revisiones son cruciales, ya que diferentes personas pueden trabajar en el mismo documento sin problemas; y también facilita la recuperación de versiones antiguas.

■ Wiki

Otra herramienta interesante son los *wikis*, que facilitan el trabajo con documentos utilizando un navegador web. La página de Wikipedia nos dice sobre los *wikis* [2]:

Un wiki (o una wiki) [...] es un sitio web colaborativo que puede ser editado por varios usuarios. Los usuarios de una wiki pueden así crear, modificar, borrar el contenido de una página web, de forma interactiva, fácil y rápida; dichas facilidades hacen de la wiki una herramienta efectiva para la escritura colaborativa.

Los *wikis* permiten trabajar en documentos a través de la web, utilizando únicamente un navegador. Está pensado para documentos breves. En nuestra opinión, no es una buena herramienta para escribir entregables, pero sí es apropiada para organizar una base de conocimiento acerca del proyecto para el equipo investigador.

■ Sistema de gestión de fallos

Finalmente, puede ser interesante utilizar un *sistema de gestión de fallos*. La Wikipedia nos da esta definición sobre estas herramientas [3]:

Un sistema de seguimiento de errores es una aplicación informática diseñada para asistir a los programadores y otras personas involucradas en el desarrollo y uso de siste-

mas informáticos en el seguimiento de los defectos de software. [...].

Uno de los componentes principales de un sistema de seguimiento de errores es la base de datos donde se almacenan los hechos e historia de un fallo de software. Los hechos pueden ser una descripción detallada del fallo, la severidad del evento, forma de reproducirlo y los programadores que intervienen en su solución así como información relacionada al proceso de administración de la corrección del fallo como puede ser personal asignado, fecha probable de remedio y código que corrige el problema.

En los proyectos de investigación, el sistema de seguimiento de fallos puede ser usado por los gestores del proyecto para asignar tareas a las personas, y para seguir la evolución del trabajo. Esto hace la vida más fácil al jefe de proyecto, y permite a todo el mundo estar al tanto del día a día del trabajo de las demás personas del grupo.

En nuestra opinión, estas son las herramientas fundamentales que deben usar los grupos que desarrollan proyectos de investigación. Esto facilita la organización del trabajo y permite tener un seguimiento bastante preciso del día a día.

3.1. Herramientas que satisfacen estas necesidades

Sitio web

Para el primer requisito, un sitio web con funcionalidad CMS, hay muchas opciones disponibles basadas en software libre. Se puede encontrar una lista de soluciones en [5]. Muchas de ellas pueden satisfacer todas las necesidades de un grupo de investigación, como la existencia de perfiles de acceso a los documentos (públicos, privados, etc). Sin embargo, nuestra recomendación no está incluida en la lista anterior. Nosotros recomendamos el uso de Plone [9].

Listas de correo electrónico

Al respecto de las listas de correo, recomendamos GNU Mailman, que es un completo paquete de gestión de listas. Tiene una interfaz web para administración, y permite archivar los mensajes y acceder al archivo mediante una interfaz web. Se puede encontrar más información en su página de la Wikipedia [4].

Sistema de control de revisiones

Para construir un repositorio de ficheros con control de revisiones recomendamos Subversion [6] (también conocido como SVN). El principal motivo es que Subversion se integra bastante bien con otras herramientas, y puede utilizarse con clientes Webdav, disponibles en muchos navegadores de archivos de casi cualquier sistema opera-

tivo. En cualquier caso, para acceder al repositorio es mejor utilizar un cliente específico de Subversion, para poder aprovechar todas sus capacidades.

Wiki

Para las wikis, en nuestra opinión la solución a elegir es el popular MediaWiki, que es el sistema utilizado por ejemplo en Wikipedia [7].

Sistema de gestión de fallos

Finalmente, para el sistema de gestión de fallos recomendamos Trac [8]. Lo más interesante de Trac es que integra, además del sistema de gestión de fallos, una wiki, un repositorio Subversion, y una línea de tiempo para la planificación de proyectos. Por ejemplo, cuando se envía un informe de fallo (llamado *ticket* en Trac), se puede asociar a un hito de planificación, a un documento mantenido en su repositorio de Subversion o a las personas involucradas en él. La información está disponible en varios formatos accesibles por web: formato texto y RSS. En particular, con la traza basada en RSS se pueden realizar seguimientos e informes de actividad. Hay, sin embargo, muchas alternativas para sistemas de gestión de fallos. En [10] encontraremos una lista extensa de sistemas de este tipo, clasificados según diferentes categorías.

4. Organización del trabajo

En esta sección presentaremos nuestra experiencia de uso de las herramientas anteriores para satisfacer las necesidades de nuestra participación en algunos proyectos europeos. En primer lugar, estas son las herramientas que seleccionamos:

- Zope, para el sitio web.
- Mailman, para las listas de correo.
- Subversion, como sistema de control de versiones.
- Trac, para el wiki y el sistema de gestión de fallos. El repositorio de versiones Subversion está integrado con Trac.

Para el sitio web, hemos desarrollado nuestra propia solución, utilizando Zope como plataforma. El sitio web no cumple con todos los requisitos (repositorio de documentos, perfiles para diferentes clases de usuarios, etc.). Sin embargo, en el ámbito de cada proyecto, sí se han adoptado herramientas que cumplen los requisitos. Por ejemplo, en algunos proyectos se utiliza Plone (un CMS que se ejecuta sobre Zope).

Para la gestión de cada proyecto, hemos determinado usar tres listas de correo:

- Una lista donde todas las personas involucradas en el proyecto pueden participar.
- Una lista donde solo los gestores del proyecto pueden participar.
- Una lista donde se suscribe a todos los

socios. Ésta es útil cuando el Trac da soporte al trabajo de un solo equipo, pero en el proyecto participan varios equipos de diferentes instituciones.

- Una lista para los seguidores de los *commits*. Cada vez que alguien envía una nueva versión de un fichero al sistema de control de revisiones, se envía automáticamente un mensaje con la descripción del envío a esta lista. Esto permite a quien se suscriba estar al tanto de todos los cambios realizados en el repositorio.

Para las listas de correo utilizamos Mailman. Las listas se configuran habitualmente como moderadas para personas no suscritas (es decir, los mensajes de personas no suscritas quedan retenidos) para evitar correo no deseado (*spam*). Algunas listas, como las destinadas a los gestores del proyecto, se pueden configurar además como privadas (es decir, nadie puede leer los mensajes ni suscribirse sin autorización).

Para la wiki y el sistema de seguimiento de fallos hemos adoptado Trac. También hemos integrado en ese Trac el repositorio de control de versiones Subversion. Utilizamos la wiki como base de conocimientos del proyecto, y el sistema de seguimiento de fallos para controlar, asignar y monitorizar el trabajo en el proyecto. Además, los mensajes de correo electrónico generados por esta herramienta (cuando se crean, modifican o cierran *tickets* del Trac) son enviados a la lista utilizada por el grupo de trabajo.

Cuando se gestionan varios proyectos simultáneamente, y cada uno cuenta con su propio Trac, es muy útil integrar el seguimiento de la actividad de todos los proyectos en un "*planeta*" (un programa de agregado de RSS¹). El *planeta* es muy útil para ver la actividad más reciente de todos los proyectos, en una sola página web, para lo que se importan todos los RSS que representan la línea de tiempo de cada uno de los Trac.

Nuestro equipo investigador ha modificado el software Planet con el fin de incluir *indicadores de actividad*. Un *indicador de actividad* es una caricatura que representa la actividad reciente de un proyecto. Por ejemplo, si un proyecto ha registrado actividad en la última hora, la caricatura corresponde a una cara que ríe. Pero cuando la actividad más reciente es del día anterior, la expresión es más seria. Hay diferentes rostros hasta llegar al peor caso, que representa a cualquier proyecto que no haya tenido actividad en el último mes.

Un ejemplo de este sitio web se encuentra en la **figura 1**. En la parte izquierda vemos una lista de eventos recientes, clasificados por proyecto. En la parte derecha, vemos una lista de todos los proyectos, con la indica-

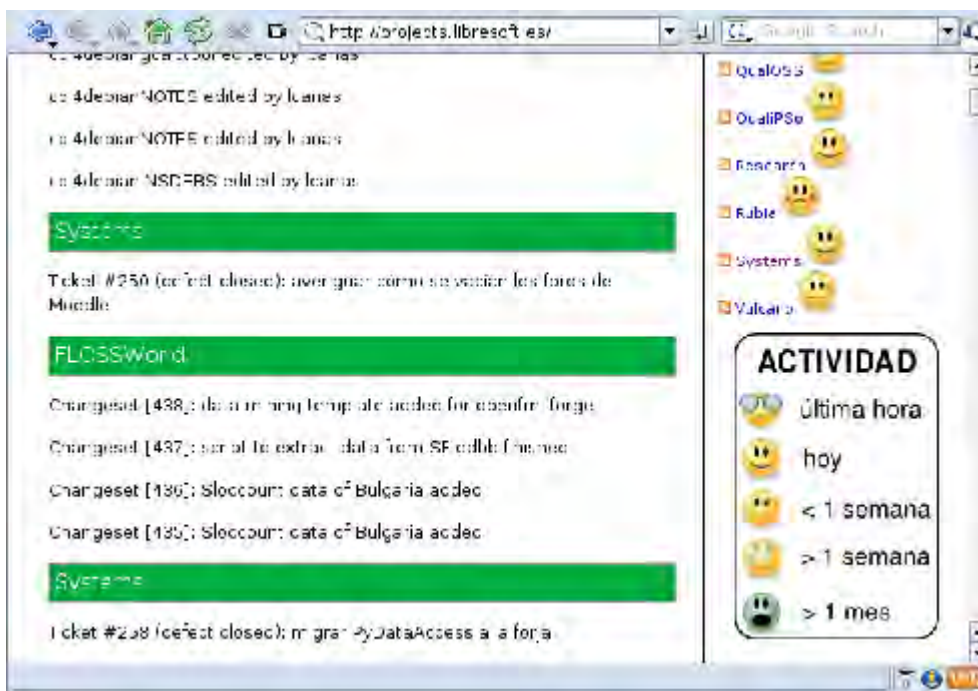


Figura 1. Planeta, con la actividad reciente de todos los proyectos, e indicadores de actividad.

ción de actividad reciente. También se incluye un resumen de los indicadores de actividad utilizados.

El fichero RSS de cada herramienta Trac es el que integramos en el sitio web llamado *planeta*. Este fichero contiene una entrada por cada evento sucedido en el Trac. Puede ser por creación o finalización de un *ticket*, por un cambio en la *wiki* (crear, modificar o borrar una página) o por un cambio registrado en el sistema Subversion asociado (adición, borrado o modificación de un fichero).

El *planeta* ha resultado muy útil para nuestro grupo. En primer lugar porque permite a cualquiera que trabaje en el grupo estar al tanto de toda la actividad reciente en todos los proyectos, y quién está participando. En segundo lugar, porque los indicadores de actividad actúan como "motivadores" de los diferentes equipos que trabajan en cada proyecto. Por ejemplo, si un grupo se pone a la cabeza de la actividad reciente, otro grupo se puede ver motivado a potenciar su actividad para recuperar el liderazgo.

En resumen, hemos puesto en producción todas las herramientas indicadas. Al integrarlas, hemos obtenido los mejores resultados de usarlas. Por ejemplo, nuestros sitios con Trac integran *wiki*, repositorios Subversion y sistemas de seguimientos de fallos. Además, tenemos listas de correo que reciben un mensaje cada vez que el Trac registra algún cambio. En un sitio web unificado podemos tener constancia de la actividad de todos los proyectos. Esto permite a cualquiera estar al tanto del trabajo reciente, realizado por el resto de grupo, independien-

temente de que trabajen en esos proyectos o no. Y además los indicadores de actividad resultan dinamizadores de la actividad de los proyectos, al comparar la actividad de unos proyectos con otros.

Sin embargo, hemos de admitir que debido a requisitos ajenos a nosotros, no podemos abrir el acceso a nuestras herramientas todavía. Por ahora estamos disfrutando de las ventajas de poder compartir el conocimiento con el resto de socios. Estamos haciendo esfuerzos para conseguir abrirlo a más usuarios.

5. Conclusiones

En este artículo, hemos presentado una metodología y una serie de herramientas, destinados a organizar el trabajo en proyectos de investigación así como a sus distintos grupos participantes. Nuestra metodología está basada en los métodos y herramientas utilizados para gestionar las comunidades de software libre.

Los proyectos de investigación deberían ser abiertos, como el software libre, por dos razones: están financiados con fondos públicos, por lo que los resultados deberían ser públicos; y porque algunos proyectos se pueden beneficiar de la colaboración con otros proyectos de investigación, haciendo más eficiente el gasto de fondos públicos.

La metodología propuesta permite que toda la información sea de libre disposición. No solamente los entregables finales sino también todo el trabajo desarrollado durante la vida del proyecto.

Las herramientas propuestas permiten ha-

cer seguimiento de todo el trabajo realizado durante la vida del proyecto. Estos repositorios de información acerca del proyecto de investigación abren nuevas direcciones para mejorar la eficacia de este tipo de proyectos. Por ejemplo, para la vigilancia tecnológica automática de los proyectos de investigación, basada en las trazas disponibles en los repositorios (sitio web, listas de correo, sistema de control de versiones, gestor de fallos, etc).

Por otro lado, las herramientas propuestas y los métodos permiten filtrar parte de la información a determinados grupos de usuarios o al público en general, basándose en determinados perfiles de niveles de acceso a la información.

Otro punto interesante de esta metodología es que hace posible trabajar de forma remota, ya que toda la información gestionada por las herramientas, así como las propias herramientas están accesibles de manera remota. Esto permite que los participantes puedan continuar trabajando incluso cuando realizan viajes a reuniones de proyecto o estancias de investigación en otros centros. Estas herramientas también permiten la coordinación entre diferentes socios, normalmente localizados en diferentes países.

El único inconveniente de nuestra propuesta es que es solo válida para proyectos de tecnologías de información y comunicaciones. Por ejemplo, para proyectos sobre química o biología, es necesario que las personas trabajen en un lugar común. En cualquier caso, las herramientas podrían ser de utilidad para gestionar el control de los entregables, por

ejemplo, aunque el teletrabajo en determinados proyectos no sea posible.

En un trabajo futuro, utilizaremos las trazas de los repositorios de los proyectos en los que trabajamos para construir una plataforma de vigilancia tecnológica de la investigación en software libre. Estamos también planeando la construcción de herramientas para automatizar los informes de actividad y participación, basándonos en la información proporcionada por los repositorios. En un futuro próximo, consideraremos también abrir todos nuestros repositorios, para poner la información a disposición del público. Por ahora, al estar trabajando con otros socios, la decisión no está en nuestras manos. En cualquier caso, todos los resultados públicos de nuestros proyectos se ofrecen mediante licencias de uso no restrictivas, tanto en el caso del software como de los documentos.

Referencias

- [1] <<http://es.wikipedia.org/wiki/Cms>>.
- [2] <<http://es.wikipedia.org/wiki/Wiki>>.
- [3] <http://es.wikipedia.org/wiki/Sistema_de_seguimiento_de_errores>.
- [4] <http://en.wikipedia.org/wiki/GNU_Mailman>.
- [5] <<http://www.opensourcecms.com>>.
- [6] <[http://en.wikipedia.org/wiki/Subversion_\(software\)](http://en.wikipedia.org/wiki/Subversion_(software))>.
- [7] <<http://en.wikipedia.org/wiki/Mediawiki>>.
- [8] <<http://en.wikipedia.org/wiki/Trac>>.
- [9] <<http://plone.org>>.
- [10] <http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems>.

Nota

¹ El más utilizado, escrito en Python, se encuentra disponible en <<http://www.planetplanet.org/>>.

NOVEDAD

Certificación CEOS (Certificación de Empresas de Open Source)

En colaboración con la agencia IQUA, la asociación CatPL, el Institute of Law and Technology de la UAB, OpenTrends, SUN Microsystems, la Universidad de Lleida, CommuniTV, Microgénesis y Applus+ LGAI.

CEOS es el referencial que busca la excelencia en desarrollos basados en software libre. Es por tanto de aplicación para empresas que desarrollan así sus productos y desean proporcionar garantía de confianza a sus clientes. Parte de estándares de calidad conocidos incidiendo en aspectos propios como el uso de entornos colaborativos, el proceso de selección de módulos, la gestión de licencias, etc.

Para más información, contactar con
Arsenio Tortajada: <atortajada@appluscorp.com>.