

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <<http://www.upgrade-cepis.org/>>

#### Publisher

UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <<http://www.cepis.org/>>) by **Novática** <<http://www.ati.es/novatica/>>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <<http://www.ati.es/>>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**

UPGRADE was created in October 2000 by CEPIS and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <<http://www.svifsi.ch/>>)

UPGRADE is the anchor point for **UPENET** (UPGRADE European Network), the network of CEPIS member societies' publications, that currently includes the following ones:

- **Informatik-Spektrum**, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- **ITNOW**, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
- **Novática**, journal from the Spanish CEPIS society ATI
- **OCG Journal**, journal from the Austrian CEPIS society OCG
- **Pliroforiki**, journal from the Cyprus CEPIS society CCS
- **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

#### Editorial Team

Chief Editor: Llorenç Pagés-Casas, Spain, <[pages@ati.es](mailto:pages@ati.es)>

Associate Editors:

François Louis Nicolet, Switzerland, <[nicolet@acm.org](mailto:nicolet@acm.org)>

Roberto Carniel, Italy, <[carniel@dgt.uniud.it](mailto:carniel@dgt.uniud.it)>

Zakaria Maamar, Arab Emirates, <[Zakaria.Maamar@zu.ac.ae](mailto:Zakaria.Maamar@zu.ac.ae)>

Soraya Kouadri Mostéfaoui, Switzerland, <[soraya.kouadrimostefaoui@gmail.com](mailto:soraya.kouadrimostefaoui@gmail.com)>

Rafael Fernández Calvo, Spain, <[rfoalvo@ati.es](mailto:rfoalvo@ati.es)>

#### Editorial Board

Prof. Wolfried Stucky, CEPIS Former President

Prof. Nello Scarabottolo, CEPIS Vice President

Fernando Piera Gómez and Llorenç Pagés-Casas, ATI (Spain)

François Louis Nicolet, SI (Switzerland)

Roberto Carniel, ALSI – Tecnoteca (Italy)

#### UPENET Advisory Board

Hermann Engesser (Informatik-Spektrum, Germany and Switzerland)

Brian Runciman (ITNOW, United Kingdom)

Franco Filippazzi (Mondo Digitale, Italy)

Llorenç Pagés-Casas (Novática, Spain)

Veith Risak (OCG Journal, Austria)

Panicos Masouras (Pliroforiki, Cyprus)

Andrzej Marciniak (Pro Dialog, Poland)

Rafael Fernández Calvo (Coordination)

**English Language Editors:** Mike Andersson, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Jim Holder, Pat Moody, Brian Robson

**Cover page** designed by Concha Arias Pérez

"Exit of Room 101" / © ATI 2007

**Layout Design:** François Louis Nicolet

**Composition:** Jorge Llácer-Gil de Rameles

**Editorial correspondence:** Llorenç Pagés-Casas <[pages@ati.es](mailto:pages@ati.es)>

**Advertising correspondence:** <[novatica@ati.es](mailto:novatica@ati.es)>

UPGRADE **Newslist** available at

<<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>>

#### Copyright

© Novática 2007 (for the monograph)

© CEPIS 2007 (for the sections UPENET and CEPIS News)

All rights reserved under otherwise stated. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, contact the Editorial Team

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (February 2008)

### "ICT Governance"

(The full schedule of UPGRADE is available at our website)



The European Journal for the Informatics Professional  
<http://www.upgrade-cepis.org>

Vol. VIII, issue No. 6, December 2007

### Monograph: Free Software: Research and Development (published jointly with Novática\*)

Guest Editors: *Manuel Palomo-Duarte, José-Rafael Rodríguez-Galván, Israel Herraiz-Tabernero, and Andrea Capiluppi*

- 2 Presentation. Free Software: Scientific and Technological Innovation — *Andrea Capiluppi, José-Rafael Rodríguez-Galván, Manuel Palomo-Duarte, and Israel Herraiz-Tabernero*
- 5 The Need for Libre Software Research in Europe — *Israel Herraiz-Tabernero, José-Rafael Rodríguez-Galván, and Manuel Palomo-Duarte*
- 8 From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects — *Andrea Capiluppi and Martin Michlmayr*
- 18 The Commons as New Economy and what this Means for Research — *Richard P. Gabriel*
- 22 Libre Software for Research — *Israel Herraiz-Tabernero, Juan-José Amor-Iglesias, and Álvaro del Castillo-San Félix*
- 27 Technological Innovation in Mobile Communications Developed with Free Software: Campus Ubicuo — *Javier Carmona-Murillo, José-Luis González-Sánchez, and Manuel Castro-Ruiz*
- 34 The Case of the University of Cádiz's Free Software Office Among Spanish Universities — *José-Rafael Rodríguez-Galván, Manuel Palomo-Duarte, Juan-Carlos González-Cerezo, Gerardo Aburruga-García, Antonio García-Domínguez, and Alejandro Álvarez-Ayllón*
- 40 On Understanding how to Introduce an Innovation to an Open Source Project — *Christopher Ozbek and Lutz Prechelt*
- 45 3D Distributed Rendering and Optimization using Free Software — *Carlos González-Morcillo, Gerhard Weiss, David Vallejo-Fernández, Luis Jiménez-Linares, and Javier Albusac-Jiménez*
- 54 Identifying Success and Tragedy of FLOSS Commons: A Preliminary Classification of Sourceforge.net Projects — *Robert English and Charles M. Schweik*

### UPENET (UPGRADE European Network)

- 60 From **Novática** (ATI, Spain )  
ICT Security  
Security of Electronic Passports — *Václav Matyáš, Zdeněk Řiha, and Petr Švéda*

\* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <<http://www.ati.es/novatica/>>.

# Libre Software for Research

*Israel Herraiz-Tabernero, Juan-José Amor-Iglesias, and Álvaro del Castillo-San Félix*

*Traditionally, research projects tend to be less than transparent, only showing to the public selected deliverables but no internal information. Normally no information about how the research project is progressing is available as public data. Even the partners of the project tend to be unaware of how the other partners are getting on. In this respect, research projects are similar to traditional software development projects. Research projects in the field of Information Society Technologies share some features with libre (free / open source) software projects, such as global distributed development and the possibility of teleworking. In the light of the above, in this paper we present a proposal to manage research projects, adopting methods used in the libre software community, and using libre software tools. Our methodology facilitates communication flows between the various partners of the project, even if they are geographically dispersed, and also allows selected internal information to be shared with the general public. Furthermore, by adopting this methodology, several additional possibilities arise, among which are automated public activity reports, project progress analyses, and technological watching and foresight techniques. We firmly believe that this new approach to managing research projects presents a number of advantages over traditional organization methods, and may improve the performance of research projects.*



Herraiz, Amor and del Castillo, 2007. This article is distributed under the “Attribution-Share Alike 2.5 Generic” Creative Commons license, available at <http://creativecommons.org/licenses/by-sa/2.5/> >.

**Keywords:** Framework Programme, Free Software, Libre Software, Open Source, Research, Research Management.

## 1 Introduction

Within the scope of the 6th Framework Programme (6FP), libre (free / open) source has begun to arouse interest, and several projects have been studying the phenomenon with a view to increasing knowledge and improving software development. Many of the good practices applied in libre software projects could be adapted to the management of complex environments. We think

that one of these complex environments could be research projects themselves.

In a research project, people from different countries work in coordination to achieve the goals of the project. These people, often in different geographical locations, need to work on the same documents or on the same pieces of software, and consequently need to be aware of the work of the other partners to ensure an efficient division of work.

Traditionally, however, research projects tend to be less than transparent. Partners are not fully aware of what

## Authors

**Israel Herraiz-Tabernero** is a Ph.D. student at the *Universidad Rey Juan Carlos*, Madrid, Spain. His research is related to the evolution of libre software projects. In particular, he is using time series analysis and other statistical methods to characterize the evolution of software projects. He has participated in several research projects funded by the Framework Programme of the European Commission (QUALOSS, FLOSSMetrics, QUALIPSO, CALIBRE). He has also collaborated on other projects funded by companies such as Vodafone and Telefonica. He has participated in the writing of manuals about managing and starting libre software projects. For example, together with Juan José Amor and Gregorio Robles he wrote a manual for the *Universitat Oberta de Catalunya*'s Master Programme in Free Software. He has been a reviewer for the IEEE Africon 2007 among other conferences and for the journal IEEE Transactions on Software Engineering. He is currently a research and teaching assistant at the *Universidad Rey Juan Carlos*, pursuing his PhD on the evolution of libre software. He also coordinates the programme of the Libre Software Master offered by the *Universidad Rey Juan Carlos*, in collaboration with Igalia and Caixa Nova. <herraiz@gsync.escert.urjc.es>.

**Juan-José Amor-Iglesias** has an M.Sc. in Computer Science from

the *Universidad Politécnica de Madrid* and he is currently pursuing a Ph.D. at the *Universidad Rey Juan Carlos*, where he is also a project manager. His research interests are related to free software engineering, mainly effort and schedule estimates in free software projects. Since 1995 he has collaborated in several free software related organizations: he is a co-founder of LuCAS, the best known free documentation portal in Spanish, and Hispalinux, and he also collaborates with [Barrapunto.com](http://Barrapunto.com) and Linux+. <jjamor@gsync.escert.urjc.es>.

**Álvaro del Castillo-San Félix** has been in Free Software communities for over 12 years. His first experience was with the Linux kernel and all his professional life has been related to Free Software projects. He has worked on the foundation of projects such as [Barrapunto.com](http://Barrapunto.com) and large communities such as GNOME Hispano and Mono Hispano. He worked as project manager for the first version of LinEx and during his role as Architectural Director in LambdaUX software company he participated in the [Compatiblelinux.org](http://Compatiblelinux.org) project. He works in the GNOME community, mainly on the Planner project where he has spent over four years coding, and also on projects like FSpot and Evolution. He is currently coordinating the European Free Software Projects in the GSyC/LibreSoft research group at the URJC university where he lectured for four years in distributed systems. <acs@gsync.escert.urjc.es>.

the rest of the partners are doing, the general public may access only selected documents, commonly referred to as deliverables, and not all deliverables are made available to the public.

This is a serious problem. First of all, at least within the scope of the 6th Framework Programme, research projects are publicly funded. Therefore all results (not only the final deliverables but all the work done in the project) should be available to those who are paying for the project.

Furthermore, at least in the case of libre software projects in the 6th Framework Programme (and probably in other fields too), several projects partially share the same goals and need access to the same sources of information. These projects could gain from other similar projects if they could access the internal documents and information generated by each project. Think of the analogy with the libre software world: if developers know that they can reuse a piece of source code available in any other project, they can simply take it and adapt it for their own purposes.

Because of all these issues we propose a methodology to adapt the practices applied in the libre software community to the management of research projects. Our methodology is intended to be adopted by all of the partners of a given project. The paper continues as follows. The next section describes the characteristics of a typical research project. Section 3 describes the needs of a research project and proposes tools to meet these needs. Section 4 explains how to organize the work and the environment of tools supporting that work, based on the experience of our research work. Finally, Section 5 draws some conclusions.

### 2 Structure of a Research Project

In this section we describe the structure of a typical research project. We take as examples our experience in research projects within the scope of the 6th Framework Programme.

Research projects are proposed and developed by a number of partners from different countries. This gives rise to the first problem we encounter when working on a project: language. English tends to be the language chosen for all communication between partners and for all internal and public documents generated.

The work is divided into *workpackages*. Each partner may lead one or more workpackages and all partners will participate in at least one workpackage. These workpackages will contain both *milestones* and *deliverables*. Milestones are key dates on which a certain piece of work is due. Deliverables are documents (although they may also be software, a database, etc) forming part of the final outcome of the project. Some deliverables are public, some internal to be used by the project partners, and others are intended to be delivered to the sponsor of the project (in 6FP's case, the European Commission).

The work required to produce the deliverables usually needs to be performed by various partners in coordination. Usually, one of the partners acts as coordinator and looks after all the economic aspects of the projects, while ensuring

that all the work to be performed by each partner is completed according to the workplan and in a timely fashion.

The key to a research project is coordination: the various partners need to coordinate with the rest of the partners and it is very important for all partners to be aware of the work performed by the others. Of course, each partner is responsible for its own work and for delivering it on time.

### 3 Needs of a Research Project

Certain tools are required if the project is to be developed as described above. Firstly we will talk about the general concepts behind what a research project needs, before going on to propose a number of libre software tools to meet those needs.

#### ■ Website

First of all, the dissemination requirements of a publicly funded project should be covered by a website. It is usual to build a content management system (CMS) to make it easier for the partners to publish documents and for the general public to access them.

From the Wikipedia page on CMS[1]:

*A content management system (CMS) is a system used to manage the content of a Web site. CMSs are deployed primarily for interactive use by a potentially large number of contributors. For example, the software for the website Wikipedia is based on a wiki, which is a particular type of content management system.*

The website should also be capable of distinguishing between public and private documents, making private documents available only to selected users (typically the partners of the project).

#### ■ Mailing list

Secondly, in order to facilitate communication between partners, a mailing list is required. Sometimes it is a good idea to set up two different mailing lists, one for all the people involved in the project and another limited to the core group members. In our opinion, there are some strategic decisions regarding the research project that should only be discussed by the core group and not by all the researchers taking part in the project.

If the group of people working together is greater than 4 or 5, it is essential to have a mailing list. Mailing lists also provide other advantages such as a record of past messages that can be useful when new members join the group to work on the project after it has started. Usually there will be two mailing lists, one for everyone involved in the project and another just for the core group. If the research group is small, it may be enough to have just one mailing list.

#### ■ Version control system

There is also a need for a repository of working docu-

ments and software (files of any kind in general) with version control capabilities. This makes it possible to recover past versions of the documents and to work on the same documents in coordination with other people. It is also a central point where anybody can find any document or file belonging to the project. This repository is not intended for the publication of deliverables but rather to help researchers work on documents in a coordinated fashion. Control version capabilities are crucial because different people work on the same document and it may be necessary to recover a past version of a document.

### ■ Wiki

Another interesting tool is the use of *wikis*, which make it possible to work on documents using a web browser. From the Wikipedia page on wikis [2]:

*A wiki is software that allows users to create, edit, and link web pages easily. Wikis are often used to create collaborative websites and to power community websites.*

Wikis allow researchers to work on documents on the web using only a web browser. It is intended for lightweight documents. In our opinion, it is not an appropriate tool for writing deliverables but it is more than adequate for organizing the research group's knowledge base.

### ■ Issue tracking system

Finally, an *issue tracking system* may also be useful. From the Wikipedia page on this subject [3]:

*An issue tracking system [...] is a computer software package that manages and maintains lists of issues, as needed by an organization. Issue tracking systems are commonly used in an organization's customer support call center to create, update, and resolve reported customer issues, or even issues reported by that organization's other employees. An issue tracking system often also contains a knowledge base containing information on each customer, resolutions to common problems, and other such data.*

In the case of a research project, the tracking system can be used by managers to assign tasks to people and other resources, and to monitor the progress of the work. This makes the life of the project manager easier and ensures that everybody is aware of the work performed by the rest of people in the group.

In our opinion, this is the basic set of tools that any group working on a research project should make use of. They make it easier to organize and monitor the group's work on a day-to-day basis. 3.1 ToolstoMeet these Needs

### ■ Website

For the first requirement (a website with CMS ca-

pabilities) there are a number of platforms available in the libre software community. A comprehensive list of libre software alternatives may be found at [5]. Most of them include the capabilities required by a research group, such as document repository with different profiles (public, private, and so on).

However, our recommendation is not included in the above mentioned list. We recommend using Plone [9].

### ■ Mailing lists

With regard to mailing lists, we recommend using *GNU Mailman* which is a package for managing electronic mailing lists. It has a web interface to administrate the system and enables messages to be archived and accessed via a web interface. More information about GNU Mailman can be found on the relevant Wikipedia page (see [4]).

### ■ Version control repository

For the version control repository we recommend *Subversion* [6] (also known as SVN). The main reason behind our choice is that Subversion integrates better with other tools and can be accessed using standard Webdav clients, supported by the file browsers of almost every operating system, although it is better to use a Subversion specific client so as to be able to make full use of its capabilities.

### ■ Wiki

For wikis, in our opinion the solution of choice is the popular MediaWiki, the system used by Wikipedia itself [7] among others.

### ■ Issue tracking system

Finally, for our issue tracking system, we recommend Trac [8]. What is even more interesting about Trac is that it can integrate a wiki, a subversion repository, an issue tracker, and a timeline for project planning. For instance, when submitting a ticket, it can be associated with a milestone in the project planning, with a given revision of a document in the SVN repository, or with the people involved in that ticket. The information is available in other web accessible formats: text format and RSS. In particular, RSS allows information to be processed automatically, which is useful for technological tracking and activity reporting systems.

There are however a great many alternatives for issue tracking systems.[10] includes a comprehensive list of tracking systems, broken down into various categories.

## 4 Organization of the Work

In this section we present how we used the tools mentioned in the previous section to meet our needs when

participating in some European projects.

First of all, this is the list of tools that we chose:

- Zope for our website.
- Mailman for the mailing lists.
- Subversion for the control version system.
- Trac for the wiki and the issue tracker. The SVN repository is integrated with Trac.

For the website, we developed our own solution, using Zope as a framework. The website does not meet the above mentioned requirements (document repository, profiles for different kind of users, etc). However, within the scope of the project, other solutions meeting these requirements were adopted. For instance, in some projects, Plone (which is based on Zope) was chosen.

In the case of mailing lists, we have three different mailing lists for each project:

- A list to which everybody working on the project is subscribed.
- A list containing only the core group managing the project.
- A list to which all partners are subscribed. This is useful when the trac only covers the work of one team but the project has several teams from different institutions working on it.

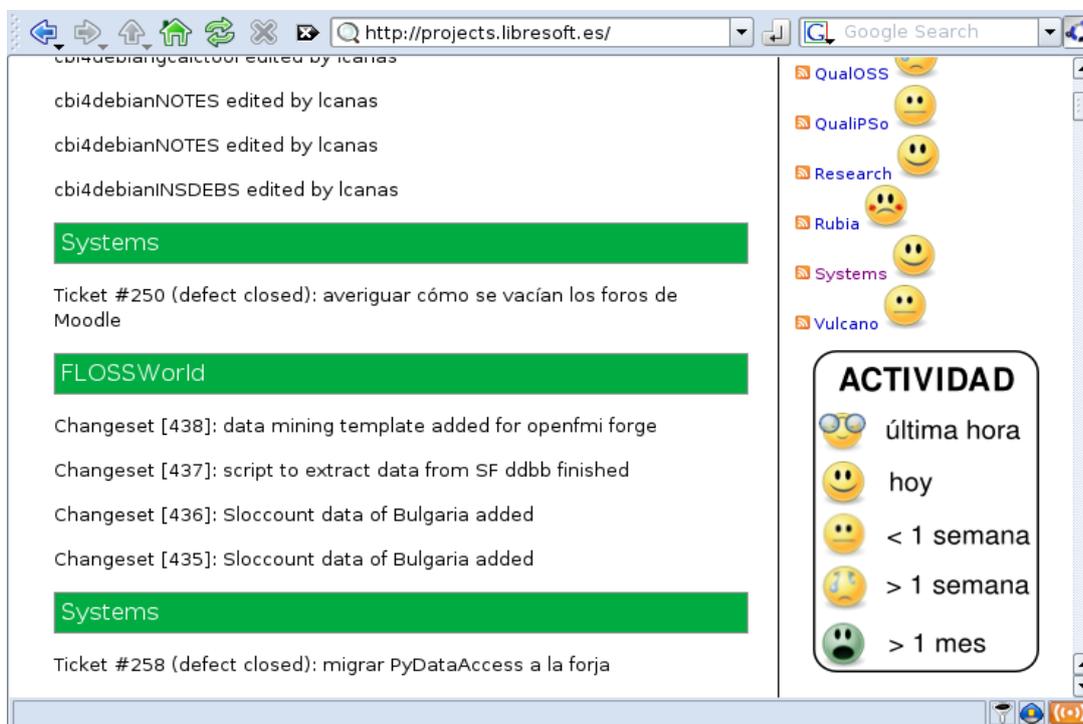
- A list of commit watchers. Every time a new commit is added to the version control system, a message with a summary of the commit is sent to this list. This allows everyone to be aware of the changes made to the repository.

For the mailing lists we use Mailman. The lists are usually configured as moderated for unsubscribed people to avoid junk emails. Some lists, such as core or partners lists, could be also configured as private (nobody can subscribe to the list or read the archives without authorization).

For the wiki and the issue tracking system, we use Trac. We also integrated the Subversion repository in Trac. We use the wiki for the project's knowledge base, and the tracking system to control, assign, and monitor the work in the project. Also, any electronic mails generated by this tool (when issue tickets are created or closed) are sent to the list used by the working team.

When managing several projects at a time, each one with its own trac, it is very useful to integrate the activity tracking of each project in a "planet" (an RSS aggregator<sup>1</sup>). Planets are very useful for seeing the recent activity of all projects in a single web page, by importing all RSS files representing the timeline content of each trac site.

Our team has modified Planet in order to integrate activity indicators as well. An activity indicator is a smiley which represents the most recent activity of a project. For



**Figure 1:** Planet Website (showing recent activity in the various projects and indicators related to this activity).

<sup>1</sup> The most commonly used, written in Python, is available at <http://www.planetplanet.org/>.

example, if a project has registered activity in the last hour, the smiley is laughing. But when the last activity is a day old, the face is more serious. There are several smileys until the worst case, representing a project which has not registered any activity in last month.

An example from this website is shown in figure 1. The left hand side displays a list of recent events, classified by project. On the right hand side, we can see a list of all the projects, with the indicator of the most recent activity. Below the list of projects there is a legend explaining the activity indicators.

The RSS feeds from the Trac tool of each project are integrated into a single website. This feed contains an entry for each event occurring in the Trac. It may be a ticket event (created, changed, etc), an event in the wiki (modification, addition or removal of a page), or an event in the Subversion repository (again modification, addition or removal).

This website has proved to be very useful for the group. First of all because it enables anyone working in the group to be aware of any recent work done in all the projects and who did it. Secondly, because the activity indicators act as a “motivator” for the various subgroups working on each project. For instance, if one group takes the lead in recent activity as shown by the indicators, another group may be encouraged to work harder to get back on top.

To sum up, we have implemented all the above mentioned tools and have realized their full potential. For example, our Trac websites integrate wiki, Subversion repository, and an issue tracking system. We also have a mailing list which receives a message every time a change occurs in any of the repositories. As we work on various projects we can consolidate the information about the recent activity of these projects in a single website. This means that everyone can be aware of the recent work performed by the rest of the group, regardless of which project they are working on. Furthermore, activity indicators act as motivators to maintain a high level of activity compared to other projects within our own research group.

However, we have to admit that due to external requirements we have not yet been able to fully open up our tools to the rest of the world. So we are not yet benefitting from sharing our knowledge with other partners working on different projects, although we are working towards that goal.

### 5 Conclusions

In this section we present a methodology and a set of tools to organize a research project and the various groups working on the project. Our methodology is based on the methods and tools used to manage and organize libre software communities.

Research projects should be as open as libre software projects are for two reasons: they are usually publicly funded and so they should be publicly available to everyone, and some projects may benefit from collaborating with other research projects, thereby making a more efficient use of public funding.

Our proposed methodology allows all information to be made publicly available. Not only the final deliverables but all the work done during the lifetime of the project.

The proposed tools make it possible to keep track of all the work performed during the entire lifetime of the project. These repositories of information on the research project open up new avenues to improve the efficiency of research projects; for example, the automated technological watching of research projects based on the trails available in the repositories of the project (website, mailing list, version control system, issue tracking, etc).

The proposed tools and methods also allow information to users and to the public to be filtered on the basis of different information access profiles.

Another strong point of this methodology is that it makes it possible to work remotely, as all the information is managed using the proposed tools and all the tools can be accessed remotely. Thus it would be possible for people visiting other partners or universities to continue working. It also enables various partners to work in coordination in spite of being in different countries.

The only drawback of our proposal is that it is only valid for Information and Communication Technologies. For instance, chemical or biological projects require people to work together at the same location. However, the tools may still be useful to organize some parts of the work, for example the management of deliverables.

In future work we will use the trails of the repositories of the projects on which we are working to build a technological watching system to track the research carried out on libre software. We are also planning to build tools to automate activity and participation reports based on the information provided by the repositories. In the near future we are also considering completely opening up our repositories to make the information available to anyone. At the moment, as we are working with other partners, that decision is not in our hands. In any event, all the results of our projects are offered under non-restrictive licenses, both for software and documents.

### References

- [1] <[http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system)>.
- [2] <<http://en.wikipedia.org/wiki/Wiki>>.
- [3] <[http://en.wikipedia.org/wiki/Issue\\_tracking\\_system](http://en.wikipedia.org/wiki/Issue_tracking_system)>.
- [4] <[http://en.wikipedia.org/wiki/GNU\\_Mailman](http://en.wikipedia.org/wiki/GNU_Mailman)>.
- [5] <<http://www.opensourcecms.com/>>.
- [6] <[http://en.wikipedia.org/wiki/Subversion\\_%28software%29](http://en.wikipedia.org/wiki/Subversion_%28software%29)>.
- [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](http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems)>.