

# Libre software for research

Israel Herraiz, Juan José Amor, Álvaro del Castillo

Grupo de Sistemas y Comunicaciones  
{herraiz, jjamor, acs}@gsyc.escet.urjc.es  
<http://libresoft.urjc.es>

**Abstract.** Traditionally, research projects are highly opaque, showing to the public only selected deliverables, but not any internal information. No information about how the research project is working is usually available as public data. Even, among the partners of the project, no information about how the rest of the partners is working, is available neither. In this sense, 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 the global distributed development, and the possibility of teleworking. Because of this reason, we present in this paper a proposal to manage research projects, adopting methods used in the libre software community, and using libre software tools. Our methodology makes possible the communication flows between the different partners of the project, even if they are in different locations, and makes also possible the sharing of selected internal information with the general public. Furthermore, by adopting these methodology, several additional possibilities arise. Among them, automated and public activity reports, evolution studies of the project, and technological surveillance and prospective techniques. We strongly think that these new approach of managing research projects present several advantages over traditional organization methods, and may boost the performance of the research projects.

## 1 Introduction

In the scope of the 6th Framework Programme (6FP), libre (free / open) source has gained attraction, and several projects have studied or are studying the phenomenon, in order to gain knowledge and improve the development of software. There are certainly good practices within libre software projects that could be adapted to manage 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 a coordinated way to get the work and the goals of the project done. These people need to work on the same documents, or on the same pieces of software, without sharing the same geographical location, and have to be aware of the work of the rest of partners, in order to be efficient in the deployment of the work.

However, traditionally, research projects are highly opaque. The partners are not fully aware of what the rest of partners is doing, the general public may access only selected documents, called *deliverables*, and even some deliverables are not intended for the public.

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

Furthermore, at least in the 6th Framework Programme's projects about libre software (and probably in other fields), several projects partially share the same goals, and need the same sources of information. These projects could gain from the other projects if they could access to the internal documents and information generated by each project. Think of the analogy with the libre software world: if a developer knows that she can reuse a piece of source code available in any other project, she can just take it and adapt for her own goal.

Because of all these issues we propose a methodology to adopt the practices found in the libre software community in the management of research projects. Our methodology is intended to be adopted by each one of the partners of the project.

The rest of the paper is as follows. Next section describes the characteristics of a typical research project. Section 3 describes the needs of a research project, and a proposal of tools that fulfill these needs. Section 4 explains how to organize the work and the environment of tools that support that work, based on the experience of our research work. Finally section 5 includes 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 the participation of research projects in the scope of the 6th Framework Programme.

Research projects are proposed and developed by a set of partners coming from different countries. This arises the first problem found when working on a project: language. English uses to be the language chosen to communicate between the partners, and to write all the documents (both internal and public) of the project.

The work is divided in *workpackages*. One of the partners may lead one or more workpackages. All the partners participate at least in a workpackage. Within these workpackages, we find both *milestones* and *deliverables*. Milestones are key dates, when some work is due. Deliverables are documents (although it can be also a piece of software, a database, etc) with a part of the final output of the project. Some deliverables are public, some internal to be used by the own partners of the project, and some other are intended to be provided to the sponsor of the project (in the case of the 6FP, the European Commission).

The work needed to get the deliverables finished must be usually done by different partners in coordination.

Usually, one of the partners acts as coordination, taking care of all the economic aspects of the projects, and making sure that all the work due by each one of the partners is finished according to the workplan and in the proper dates.

The key aspect of a research project is coordination: the different partners have to work in coordination with the rest of partners, and it is very important that all the partners are aware of the work done by the others. Of course, each one of the partners must be aware of the work done by it, and the dates when the work is due.

### 3 Needs of a research project

To get the aforementioned work done, some tools are needed by the project. We first talk about generic concepts of what a research project needs, and then we include a proposal of libre software tools to cover those needs.

#### – Website

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

From the Wikipedia page about CMS [1]:

A content management system (CMS) is a computer software system used to assist its users in the process of content management. CMS facilitates the organization, control, and publication of a large body of documents and other content, such as images and multimedia resources. A CMS often facilitates the collaborative creation of documents. A web content management system is a content management system with additional features to ease the tasks required to publish web content to Web sites.

This website should allow to distinguish among public and private documents, making private documents available only for selected users (typically, the partners of the project).

#### – Mailing list

Secondly, in order to communicate with other partners, a mailing list is needed. It is a good idea to set up two different mailing lists. One of the mailing lists would include all the people involved in the project. The other would include only the core of the group working on the project. In our opinion, some strategic decisions regarding the research project should only be discussed by a core group and not by anyone participating in the project.

If the group of people who work together is greater than 4 or 5, the need for a mailing list is clear. Furthermore, mailing lists provide other advantages such as archives of past messages, that can be useful when new members come to the group to work in the project once it has started. In this case, it

would be also a good idea to set up two mailing lists, and including all the people involved in the project and other one only with the core group. If the research group is small, it may be enough with only one mailing list.

– **Control version system**

Another need is a repository of working documents and software (files of any kind in general), with version control capabilities. This make possible to recover past version of the documents and to work in coordination with other people in the same documents. It is also a central point where anybody can find any document or file belonging to the project. This repository is not intended to publish the documents, but for the working process on the documents. The 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 are *wikis*, that may possible to work on documents using a web browser. From the Wikipedia page about wikis [2]:

A wiki [...] is a website that allows the visitors themselves to easily add, remove, and otherwise edit and change available content, typically without the need for registration. This ease of interaction and operation makes a wiki an effective tool for mass collaborative authoring.

Wikis allow to work on documents on the web, using only a web browser. It is intended for lightweight documents. In our opinion, it is not a good tool to write the deliverables, but it is good for the knowledge base of all the know-how of the research group.

– **Issue tracking system**

Finally, a *issue tracking system* maybe also useful. From the Wikipedia page about this topic [3]:

Issue tracking systems [...] are computer software packages that manage and maintain 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 others 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 the managers to assign tasks to people and other resources, and to follow the evolution of the work. This makes easier the life of the manager of the project, and allow to everybody to be aware of the work made by the rest of people in the group.

In our opinion, this is the fundamental set of tools that any group working on a research project should install. It makes easier to organize the work of the group, and it allows to keep a track of all the work done until the present day.

### 3.1 Tools to cover these needs

#### – Website

For the first requirement, a website with CMS capabilities, there exist lots 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 needed by a research group, such as document repository with different profiles (public, private and so on).

However, our recommendation is not included in that site. We recommend to use Plone [9].

#### – Mailing lists

Regarding mailing lists, we recommend to use *GNU Mailman*, which is a package for managing electronic mailing lists. It has a web interface to administrate the system, and offers the capability of archives of the list, with a web interface. More information about GNU Mailman can be found on its Wikipedia's page (see [4]).

#### – Control version repository

For the version control repository we recommend *Subversion* [6] (also known as SVN). The main reason is that Subversion has better integration with other tools, and can be accessed using standard Webdav clients, available in file navigators of almost all operating systems, although its better to use a Subversion client, which full support of all capabilities.

#### – Wiki

For wikis, in our opinion the most popular solution is MediaWiki. For instance, it is the system used by Wikipedia itself [7].

#### – Issue tracking system

Finally, for issue tracking systems, we recommend Trac [8]. What is even more interesting about Trac is that it can integrate a wiki, a subversion repository, a issue tracker, and a timeline for the planning of the project. For instance, when submitting a ticket, it can be associated to milestone in the planning of the project, to the different people participating in the project, to a given revision of a document in the SVN repository. The information is available in different formats besides web: text format and RSS. In particular, RSS allows an automatic treatment of the information, useful for technological surveillance and activity report systems.

There are however lots of alternatives for issue tracking systems. [10] include a comprehensive list of tracking systems, classified according to different categories.

## 4 Organization of the work

In this section we present how we used the mentioned tools in the previous section to cover the needs of our research projects when participating in some European projects.

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

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

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

- A list where everybody working on the project is subscribed to.
- A list containing only the core group, who manages the project.
- A list where all partners all subscribed. This is useful when the trac covers only the work of a team, but the project has several teams from different institutions working on the project.
- A list of commit watchers. Every time a new commit is made to the version control system, a message with a summary of the commit is sent to this list. This allow everyone to be aware of the changes made to the repository.

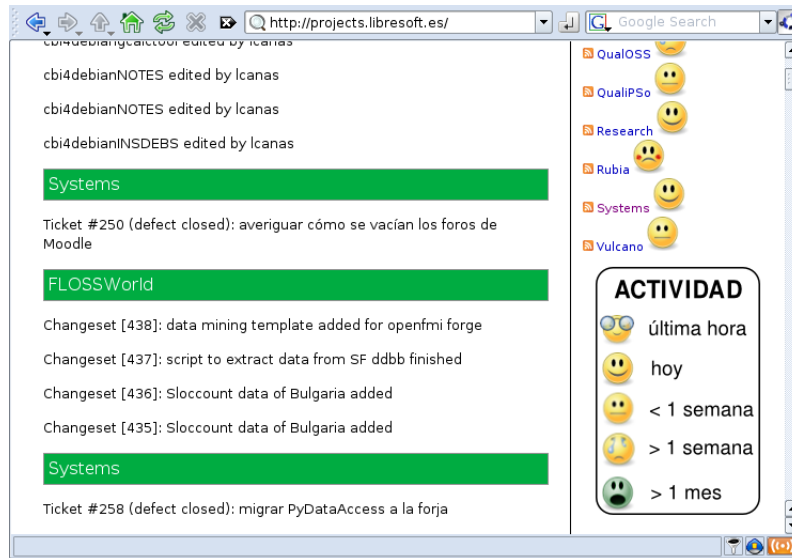
For the mailing lists we use Mailman. The lists are usually configured as moderated for unsubscribed people, to avoid junk emails. Some lists, such as core or partners lists, could be also configured as private (nobody can subscribe or read the archives without authorization).

For the wiki and the issue tracking system, we adopted Trac. We integrated also the Subversion repository in Trac. We use the wiki for the knowledge base of the project, and the tracking to control, assign and monitor the work in the project. Also, the 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 all projects in a “planet” (a RSS aggregator <sup>1</sup>). The planet is very useful to see the recent activity of all projects in a web page, importing all RSS files which represent the timeline contents of each trac site.

Our team has modified Planet in order to integrate also *activity indicators*. An activity indicator is a funny face which represents the recent activity of a

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



**Fig. 1.** Planet website, with recent activity in the different projects, and indicators about this activity

project. For example, if a project has registered activity in last hour, the face is showed laughing. But when the activity is registered only in last day, the face is showed more serious. There are several faces until the worst case, which represents the project when it has not registered activity in last month.

An example of this website is shown in figure 1. On the left part we find a list of recent events, classified by project. On the right part, we see a list of all the projects, with the indicator of the recent activity. Below the list of projects, a summary of activity indicators is shown.

The RSS feed from the Trac tool from different projects are integrated into one only website. This feed contains a entry for each event happening in the Trac. An even may be a ticket event (created, changed, etc), an event in the wiki (modification, adding or removal of a page), or an event in the Subversion repository (again modification, adding or removal).

This website has resulted to be very useful for the group. First of all because it allows to anyone working in the group to be aware of the recent work done in all the projects, and who did it. Secondly, because the activity indicators act as a “motivator” for the different subgroups working on each project. For instance, if one of the project is taking the lead in the activity indicators, other group may want to encourage their work in order to recover the leadership.

Summarizing, we have implemented all the mentioned tools. We obtained the full potential of using these tools when we integrated all of them. For instance, our Trac websites integrate wiki, Subversion repository and issue tracking system. Moreover, we have a mailing list which receives a message every time

a change happens in any of the repositories. As we are working on different projects, we join the information about the recent activity of these projects in one only website. This allow to everybody to be aware of the recent work done by the rest of the group, regardless which project they are working on. Furthermore, activity indicators resulted to be motivators to keep the activity level high, when comparing to other projects within our own research group.

However, we have to admit that due to external requirements, we can not fully open our tools to the rest of the world yet. So we are not taking advantage of sharing our knowledge with the rest of partners of the different projects. We are making efforts to achieve so though.

## 5 Conclusions

We present in this paper a methodology and a set of tools to organize a research project and the different 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 project, because of two reasons: they are usually publicly funded and so they should publicly available to anyone, and some projects may get benefit of collaborating with other research projects, making a more efficient use of the public funding.

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

The proposed tools allow to keep a track of all the work done during the whole lifetime of the project. These repositories of information about the research project open new directions to improve the efficiency of the research projects. For instance, automatic technological surveillance of research projects, based on the trails available in the repositories of the project (website, mailing list, version control system, issue tracking, etc).

In the other hand, the proposed tools and methods also allow to filter the information to the users and the public based on different profiles to access the information as well.

Another strong point of this methodology is that it makes possible to work remotely. As all the information is managed using the proposed tools, and all the tools can be available remotely. This would make possible to continue working when people is visiting other partners or universities. It allows also the coordinated work among different partners (they are usually located in different countries).

The only drawback of our proposal is that is only valid for Information and Communication Technologies. For instance, chemical or biological projects require people to work together in the same location. Although the tools may be used to organize the work anyway.

In further work we will use the trails of the repositories of the projects where we are working on to build a platform of technological surveillance of the research



done on libre software. We are planning also to build tools to automate activity and participation reports, based on the information provided by the repositories. In the near future, we are considering also to fully open our repositories, to make the information available to anyone. Right now, as we are working with other partners, that decision is in our hand. Anyway, 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\\_\(software\)](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](http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems)