

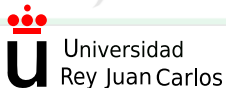
Libre Software for Research

Methodology at GSyc/Libresoft

Israel Herraiz / Juanjo Amor / Alvaro del Castillo

{herraiz,jjamor,acs}@gsyc.escet.urjc.es
GSyc/Libresoft

March 8th 2007



GSyC

LibreSoft

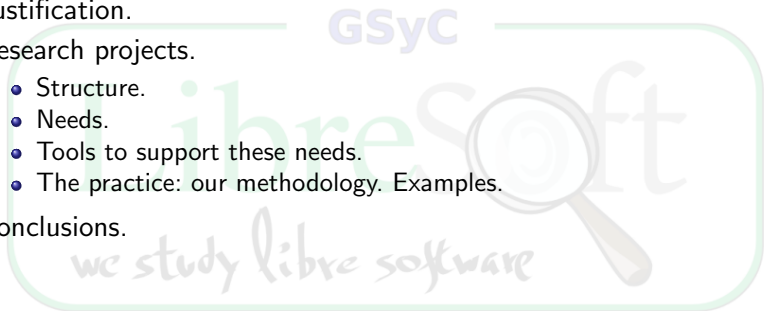
(cc) 2007 Israel Herraiz, Juanjo Amor, Alvaro del Castillo

This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.1/> or send a letter to Creative Commons, 559 Nathan Abbott

Way, Stanford, California 94305, USA.

Summary

- Justification.
- Research projects.
 - Structure.
 - Needs.
 - Tools to support these needs.
 - The practice: our methodology. Examples.
- Conclusions.



Introduction

Research projects:

- Methodologies usually opaque.
- We don't know how project is done.
- Research projects similar to traditional software development projects.
- What if we apply the some Libre Software (FLOSS) methodologies to research projects?

We think that a “FLOSS approach” in research projects could be very useful and justifiable.

Justification

In the scope of 6th Framework Programme:

- People from different countries must coordinate, working over the same documents, the same pieces of software and so on.
- Some results must be public deliverables.
- FLOSS-related projects are gaining interest.

Why to use a “FLOSS methodology”?

- We could use tools to coordinate the distributed work.
- We could use these tools to maintain some results as visible to the public.
- Some “ethical” reasons: These research projects are publicly funded. So, ideally all results (not only public deliverables) would be also public. . .

Structure of the projects

- Partners from (usually) several EU countries. One of them leads the project, acting as coordinator and maintaining the communication with the Commission.
- Work divided in *workpackages*. All partners participate in, at least, one of them. Workpackages are leaded also by one partner.
- Each workpackage has *milestones* (key dates when some work is due) and *deliverables* (documents, but also could be pieces of software or complete built systems).
- Some deliverables are given to the public, and some are used internally by partners and the Commission.

Needs for the projects

Some common needs we have found in our research projects (we are computer researchers!):

- **Website.** Results are usually published through website. Sometimes, using a CMS.
- **Mailing lists.** Basic for partners communication, since they are geographically distributed.
- **SCM.** Version control systems are desirable for coordinating document creation between partners.
- **Wikis.** Sometimes, collaborative edition is done through a wiki. Also, the wikis are useful for sharing informal ideas between partners.
- **Issue tracking systems.** Tracking systems could be used for assign specific tasks to people and follow the evolution of the work.

Tools to support these needs

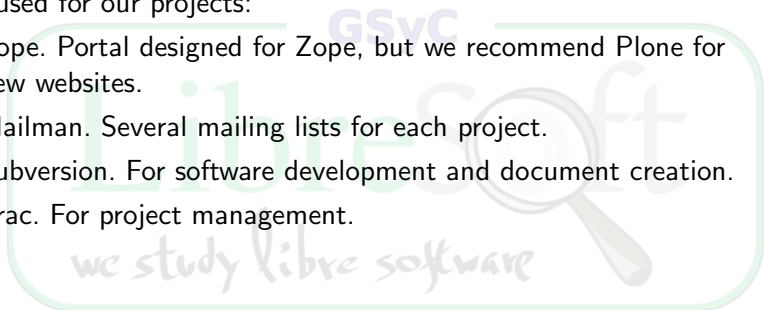
- **Website.** With CMS capabilities. There are a lot of alternatives, but we usually build websites using Plone or directly over Zope.
- **Mailing lists.** Currently, the most advanced FLOSS mailing list manager is GNU Mailman.
- **SCM.** CVS and Subversion are recommended. Subversion is more advanced, and is going to replace CVS sites.
- **Wiki.** Popular wiki systems such as MediaWiki are recommended. However, sometimes light systems such as Moin could be recommended.
- **Issue tracking systems.** There are popular issue trackers, which are easy to configure and use, such as Bugzilla.

But for our needs, integration of tools is preferable.

Our methodology

Tools used for our projects:

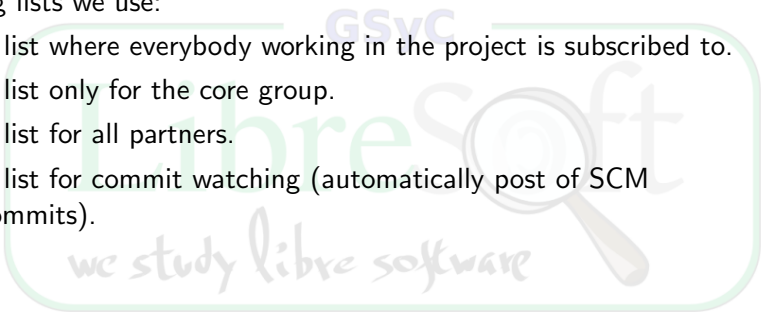
- Zope. Portal designed for Zope, but we recommend Plone for new websites.
- Mailman. Several mailing lists for each project.
- Subversion. For software development and document creation.
- Trac. For project management.



Mailing lists

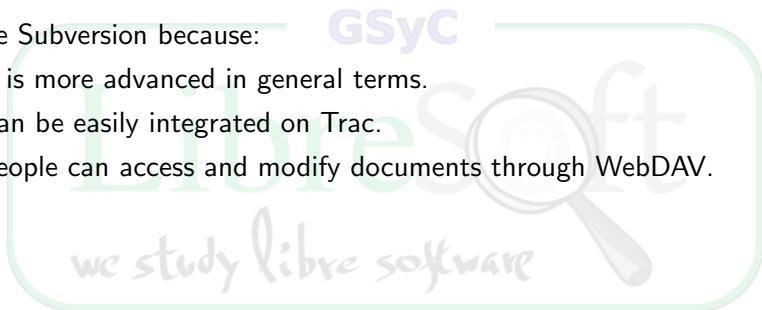
Mailing lists we use:

- A list where everybody working in the project is subscribed to.
- A list only for the core group.
- A list for all partners.
- A list for commit watching (automatically post of SCM commits).



We use Subversion because:

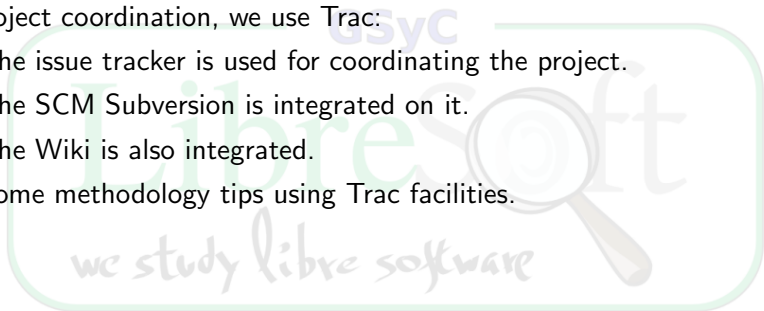
- It is more advanced in general terms.
- Can be easily integrated on Trac.
- People can access and modify documents through WebDAV.



Trac

For project coordination, we use Trac:

- The issue tracker is used for coordinating the project.
- The SCM Subversion is integrated on it.
- The Wiki is also integrated.
- Some methodology tips using Trac facilities.



Methodology: Weekly schedule

We have developed some small scripts to create on Trac Wiki:

- An index page, with links to all weeks scheduled in the project.
- A page for each week scheduled.

Each schedule page contains:

- Each involved participant.
- For each participant, a list of tasks for that week, with links to their tickets.

Methodology tips:

- The Project Manager prepares the week schedule.
- In a meeting, participants review and discuss their tasks.
- The participants can edit the wiki page to annotate their progress in their tasks.

When we work in several projects...

When we work in several projects, we use a webpage for showing “activity indicators”. Then, the Project Director can take a look to all projects and see their progress.

- Trac exports as RSS, all modifications in wiki pages, commits in SVN and ticket creations and deletions.
- We have modified a RSS aggregator (Planet) for accessing these RSS for all projects.
- The Planet shows how progress is done in each project. Also, a set of “funny faces” show us the activity level of the project.

Conclusions

Results:

- Efficient project management.
- Project coordination between distributed partners and developers.
- Easy publication of deliverables. If we don't use a separate website, we always could use the Trac wiki with links to SVN files for it.
- Easy administration: new project means a new Trac and SVN space, created in minutes.

Problems:

- Trac permissions. We cannot put a Wiki page under restricted access and rest for the public.
- Valid for Information and Communication Technologies. For other projects, not all advantages are available.
- Sometimes, it's difficult to convince partners to use the tools.

Libre Software for Research

Methodology at GSyc/Libresoft

Israel Herraiz / Juanjo Amor / Alvaro del Castillo

{herraiz,jjamor,acs}@gsyc.escet.urjc.es
GSyc/Libresoft

March 8th 2007

