
	<b>TRAC Analysis</b>	Fecha: 14/01/2009
Proyecto: TWINS		Autor: MAC
Página: 1 de 11		Revisión: 14/01/2009

## ÍNDICE

1.1.Introduction.....	2
1.1.Object.....	2
1.2.Scope.....	2
1.3.Need of a source version control.....	2
1.4.Previous situation.....	2
2.2.Analysis of SVN.....	4
2.1.MKS.....	4
2.2.Basics of Subversion.....	4
2.3.Comparison between MKS and SVN.....	7
3.3.Tools.....	9
3.1.Server.....	9
3.2.Client.....	9
4.4.SVN migration.....	10
4.1.Cvs2svn.....	10
5.5.Conclusion.....	11
6.6.Registro de Revisiones.....	12

	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Project:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 2 de 11		<b>Revisión:</b> 14/01/2009

# 1. Introduction

## 1.1. Object

The object of the present document is presenting an analysis of the use of the project manager + bug tracking application called subversion or SVN. Experiences of its use over the last years will be exposed, as well as suggestions for improvements.

## 1.2. Scope


The main purpose of the TWINS project in ZIV was to apply TWINS's techniques and tools in one project used as a pilot. However, as the use of TRAC has been such satisfactory in it, it has been extended to a full line of projects.

## 1.3. Need of a project management solution

Inside ZIV there is no standardized tool for project management. Microsoft Project is widely used to make initial Gantt charts for the project, and the efforts tracking is done with an application called B-Kin, or with Excel.

However different methods for management of tasks, improvements and bug solutions have been tried. Among all of these, Bugzilla was tested, but its use did not spread.

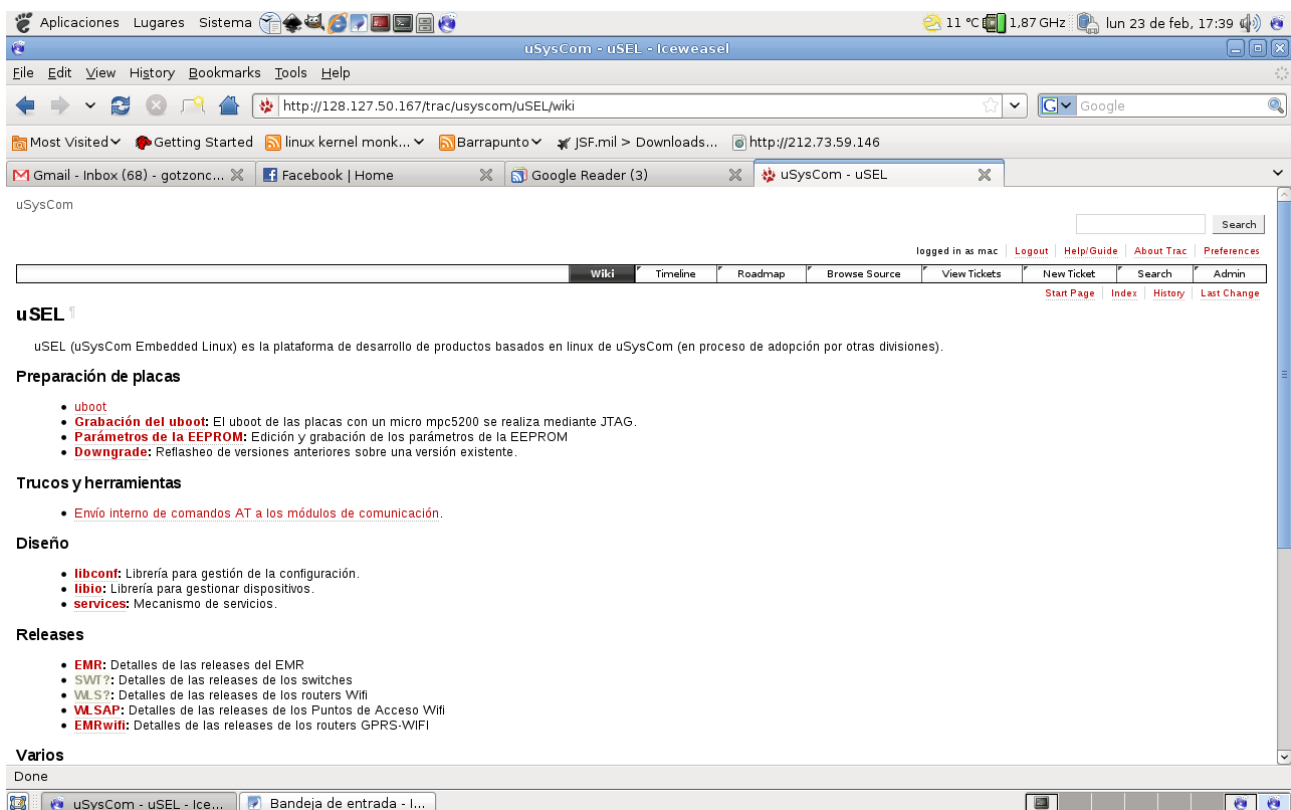
There is not neither an standardized tool for requirement description

	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Proyct:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 3 de 11		<b>Revisión:</b> 14/01/2009

## 2. Analysis of TRAC

### 2.1. Basics of TRAC


Trac is an open source, web-based project management and bug-tracking tool. The program is inspired by CVSTrac, and was originally named *svntrac* due to its ability to interface with Subversion It is developed and maintained by Edgewall Software.



From its definition, we can see that TRAC offers different tools integrated in the same solution.

#### 2.1.1 Wiki

We have found it quite useful to store in it the day-to-day information that does not match with an structured document.

	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Project:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 4 de 11		<b>Revisión:</b> 14/01/2009

In fact, application engineers use it to document the usual problems in installations...

Some other valuable information such as links to webs with information related to the project or even links to network drives is also well placed in the wiki.

One of the main advantages of using a wiki is that with it it is easy to link the same information from different points, helping in offering different views to users with different profiles.

## 2.1.2 Timeline

Trac's timeline indexes all changes done in the project. We have found that it is a good way to analyse a project subjectively (see who is actively working on it, detect bad practices of under or overcommitting...

But one of its main use is the way in which this timeline makes consulting commits easier than the command line "svn log", or diffs easier than "svn diff". So... in some aspects it has been a way to offer a gui to svn command line users.

## 2.1.3 Roadmap

Roadmap allows to combine different tickets, and set important points in the development of a project.


We have found that this is a good way to match deliverables with releases and a good companion to Microsoft Project's gantts.

## 2.1.4 Browse Source

TRAC allows the integration of an SVN repository. This has also great advantages for us:

Developers have a gui that allows to navigate inside the sources of the repository.

Non typical SVN users (managers, marketing...) can access documentation and other files stored in an SVN repository using nothing else than a web browser.

	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Proyecto:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 5 de 11		<b>Revisión:</b> 14/01/2009

## 2.1.5 Ticket management


Although the previously mentioned features are good complements, the ticket manager is the most important part of TRAC for ZIV.

Ticket manager has become a powerful helper tool in project management. The reasons are many:

- Requirements can be (and what is most important, must be) structured in well delimited descriptions.
- All developments required for a project can be expressed using tickets: Tasks / bugs / enhancements.
- The ticket manager offers a very simple workflow manager. Tickets are created, assigned, fixed, approved...
- Tickets can be grouped and assigned to an specific milestone in the roadmap.
- Information describing the process of fixing a ticket can be annexed so that a task can be fulfilled by a different developer if the first one is not capable.
- Using TRAC, SVN commits can reference TRAC tickets and TRAC tickets can reference SVN commits. This improves the integration of requirements, tests and code development, and reduces the duplicity of comments...
- Ticket manager is quite easy so noone (developers, marketing, project managers, testers...) is daunted by its use.

## 2.1.6 Search

The searching mechanism of TRAC allows to look for the information required in all sections of TRAC: wiki, timeline, tickets... This reduces the amount of time looking for things not in the right place.


	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Proyecto:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 6 de 11		<b>Revisión:</b> 14/01/2009

## 2.1.7 Security

Although TRAC can be run as an independent daemon, it can also be integrated with web servers such as Apache.

This (among other benefits), increases the security of the system, adding a first authentication point for users.

In addition, TRAC can be easily configured to create different profiles of users enforcing the use of different security politics for developers, ticket management tool users, guests...


	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Proyecto:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 7 de 11		<b>Revisión:</b> 14/01/2009

### 3. Tools

TRAC is a very simple system. Users do not need any special application to use it; their web browser is enough.

So... The only software that has to be installed and maintained is the TRAC server. As we have stated before, TRAC server can be integrated with web servers (such as Apache), due to the fact that it is programmed using Python.


In ZIV, we decided to use the same HP server that was used for SVN repositoring to install an Apache and TRAC. As this server runs a “Debian testing” system, and TRAC is integrated in Debian repositories, the installation is quick and easy. The integration with Apache is not difficult neither, and it is very well documented in the manuals and in the Internet.

	<p style="text-align: center;"><b>TRAC Analysis</b></p>	<p>Fecha: 14/01/2009</p>
<p>Proyecto: TWINS</p>		<p>Autor: MAC</p>
<p>Página: 8 de 11</p>		<p>Revisión: 14/01/2009</p>

## 4. TRAC migration

Although we did not have much valuable information in Bugzilla server, we took a look to see if there is an easy way to migrate Bugzilla tickets to SVN tickets. There is a python script (bugzilla2trac.py) that made the work easily for us.

There is no need for further migration; TRAC accesses directly to SVN repository, so it does not matter if the sVN repository is older than the TRAC system... The access to even the older commits is granted.

	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Project:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 9 de 11		<b>Revisión:</b> 14/01/2009


## 5. Problems of TRAC

Although we have found that TRAC offers us much more than we expected in the first moment, TRAC has also some lacks that prevent us from substituting other applications (such as Microsoft Project) with it.

- No multi-project: There seems no way to manage different projects with one TRAC instance. We have tried some workarounds such as creating a main page that links the different TRAC pages, or manage different projects as components inside a larger project. However it is not perfect.
- No effort management: There is no way to assign an effort to a task, so we have to map back the tasks into the Microsoft Project.
- No linked tasks: Tickets can only be grouped in a milestone inside the roadmap, but there is no way to link the start of a task with the ending of another, so there is no way to manage Gantt diagrams.
- No support for distributed VCS: We are thinking on substituting SVN with a distributed VCS such as Bazaar, Mercurial or GIT. The lack of support for these in TRAC is a problem.

### 5.1. Alternatives


Lately, there seems to be a good alternative to TRAC. Its name is Redmine, and follows most of the philosophy of TRAC solving some of the lacks stated above. The main problem is that it is programmed on Ruby on Rails, and there is not a Debian package to install it easily

	<b>TRAC Analysis</b>	<b>Fecha:</b> 14/01/2009
<b>Project:</b> TWINS		<b>Autor:</b> MAC
<b>Página:</b> 10 de 11		<b>Revisión:</b> 14/01/2009

## 6. Conclusion

TRAC has become a revolution in the way user requirements and project management for some projects in ZIV. However although it is a good companion for other applications, cannot substitute them as the only tool for project management yet.

TRAC, due to its easy of use and flexibility is helping in the collection of multidisciplinary user requirements, helping in the reduction of the gap between software and hardware engineers, and the gap between both those and other departments of the company. It is becoming a good tool to improve the HW-SW codesign.

	<b>TRAC Analysis</b>	Fecha: 14/01/2009
Proyct: TWINS		Autor: MAC
Página: 11 de 11		Revisión: 14/01/2009

## 7. Revision register

Revision	Modification	Author	Rev.	Aprov.
V1.0	Original version	MAC	MAC	