Management of astronomical software projects with open source tools.

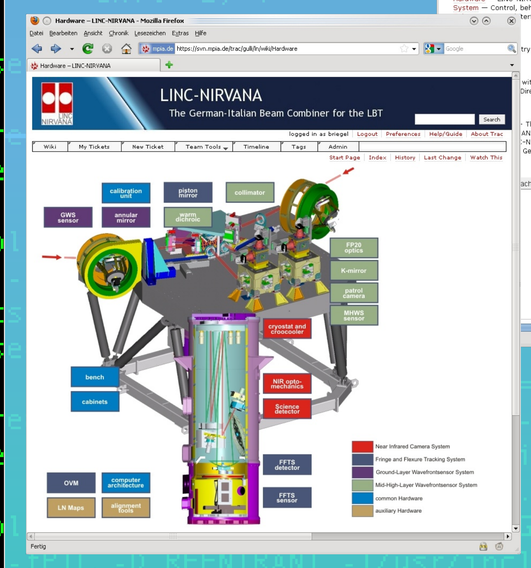
Florian Briegel[1], Thomas Bertram[1], Jürgen Berwein[1], Frank Kittmann[12]
[1] Max Planck Institute f. Astronomy, Königstuhl 17, 69117 Heidelberg, Germany
[2] University of Cologne, Zülpicher Str. 77, 50937 Cologne, Germany

Abstract

In this poster we will offer an innovative approach to managing the software development process with free open source tools, for building and automated testing [1], a system to automate the compile/test cycle on a variety of platforms to validate code changes [6], using virtualization to compile in parallel on various operating system platforms [2], version control and change management [3], enhanced wiki and issue tracking system for online documentation and reporting [4] and groupware tools as they are: blog, discussion and calendar [5].
Initialy starting with the Linc-Nirvana instrument a new project and configuration managment tool for developing astronomical software was looked for. After evaluation of various systems of this kind, we are satisfied with the selection we are using now.
Following the lead of Linc-Nirvana most of the other software projects at the MPIA are using it now.

1. Telling the Story - Instrument Documentation

Usally every project runs through a preliminary design review and afterwards a final design review, a lot of documents are produced, printed as pdf and bastet. The problem is now to find the information, mostly it is distributed in dozens of documents and it is difficult to search in pdf file. Another problem is that over the time it is realized that the final design was not that final, but the documents dont get updated regulary.



Trac has a built-in wiki engine, used for documentation throughout the system. WikiFormatting is used in wiki pages, tickets and other optional plugins. This allows for formatted text and hyperlinks in and between all Trac modules.
Editing wiki text is easy, using any web browser and a simple formatting system or a javascript based wysiwyg editor, rather than more complex markup languages like HTML, too complicated to allow fast-paced editing, and distracts from the actual content of the pages.
The main goal of the wiki is to make editing text easier and encourage people to contribute and annotate text content for a project. The wiki itself does not enforce any structure, but rather resembles a stack of empty sheets of paper, where you can organize information and documentation as you see fit, and later reorganize if necessary.

2. A plan for Action - Working with Tickets, Schedules and Milenstones.

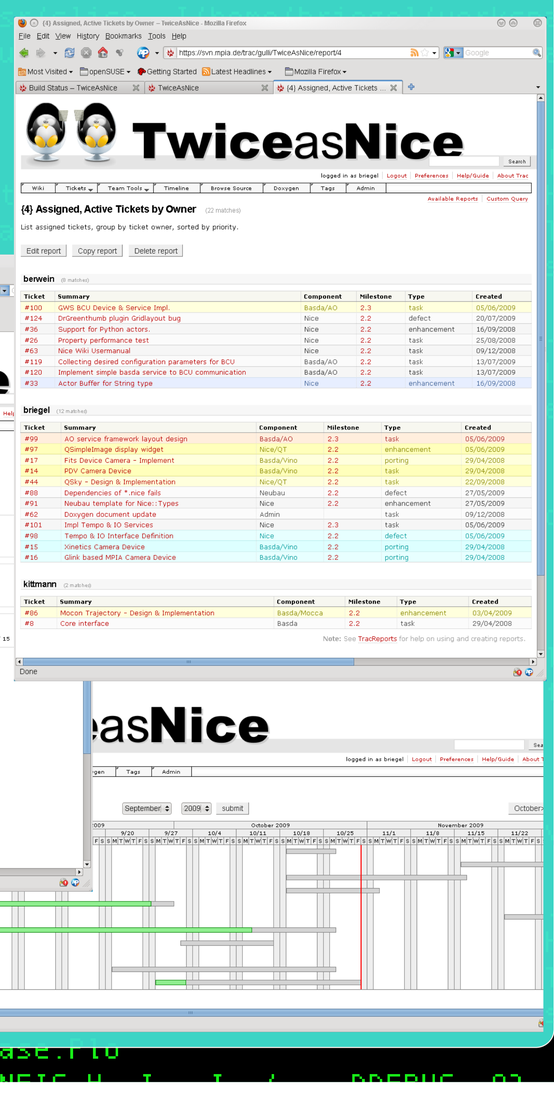
Trac Tickets - The ticket database provides simple but effective tracking of issues and bugs within a project. As the central project management element of Trac, tickets are used for project tasks, feature requests, bug reports and software support issues.

Trac Roadmap - a view on the ticket system that helps planning and managing the future development of a project.

Trac Reports - provides a simple, yet powerful reporting facility to present information about tickets in the Trac database. Rather than have its own report definition format, TracReports relies on standard SQL SELECT statements for custom report definition.

Trac Gantt Plugin - is very easy to use reporting tool for assigning time spans for completion of tickets.

Trac Blog Plugin - is a blogging system for Trac, neat thing for writing weekly reports or meetings summarisation.



3. Make it so - Working on the Software

Build System

Historically, developers used build automation to call compilers and linkers from inside a build script versus attempting to make the compiler calls from the command line. It is simple to use the command line to pass a single source module to a compiler and then to a linker to create the final deployable object. However, when attempting to compile and link hundrets source code modules, in a particular order, using the command line process is not a reasonable solution. GNU Autoools[1] and our friend neubau offer a better alternative.

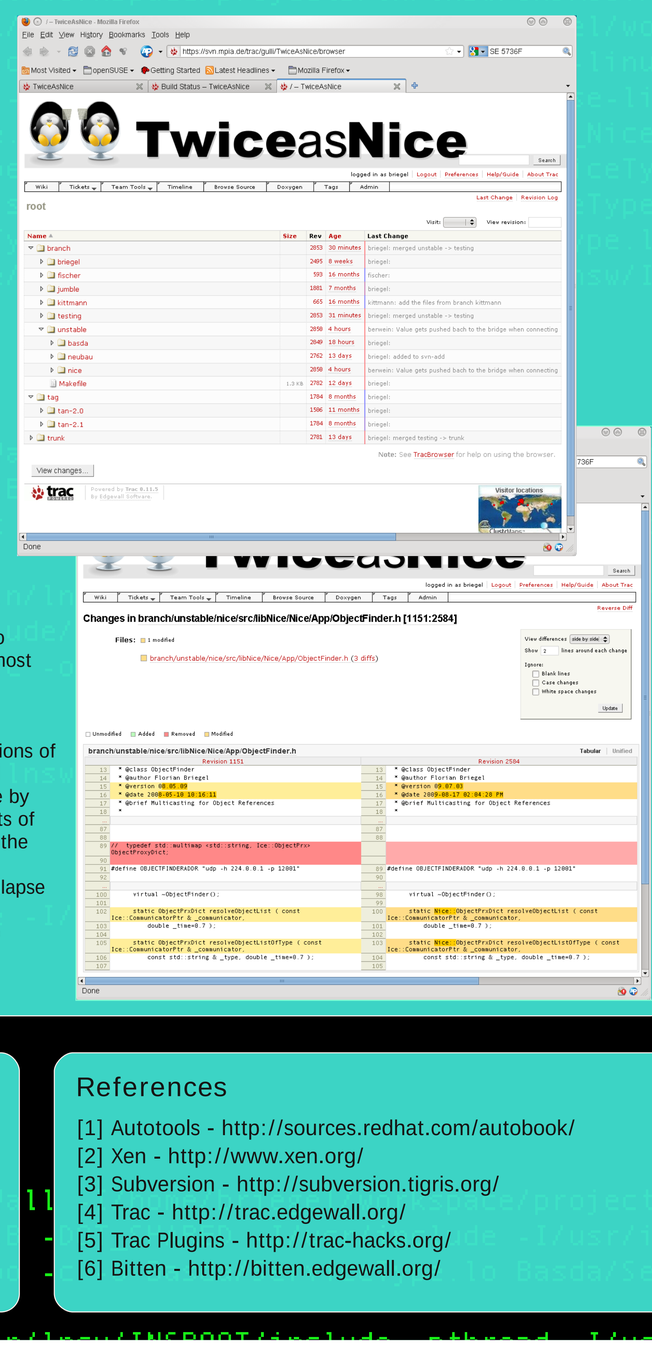
Autotools - the GNU build system, is a suite of programming tools produced by the GNU project. These tools are designed to assist in making various source code packages portable to many Unix-like systems and also Windows. It also supports automated testing with simple test programs or unit tests, installing & deinstalling, creating distribution packages.

Neubau - is a small utility to create complete autotools projects, with ready to use autoconf configuration scripts and automake build rules for various libraries, e.g. Nokia's QT widget, ZeroC's Ice and building RPM & debian packages.

Revision control

Subversion - our revision control system for the management of changes to documents, programs, and other information stored as computer files. It is most commonly used in software development, where a team of people may be changing the same files.

Trac Browser - the repository browser can be used to browse specific revisions of directories and files stored in the repository of the configured version control system. The browser can be used to navigate through the directory structure by clicking on the directory names. Clicking on a file name will show the contents of the file. Clicking on the revision number of a file or directory will take you to the TracRevisionLog for that file.
If you're using a Javascript enabled browser, you'll be able to expand and collapse folders in-place by clicking on the arrow head at the right side of a folder.
It also supports inspecting changes between different revisions with just two simple clicks.



4. The Good, the Bad & the Ugly - Test and release

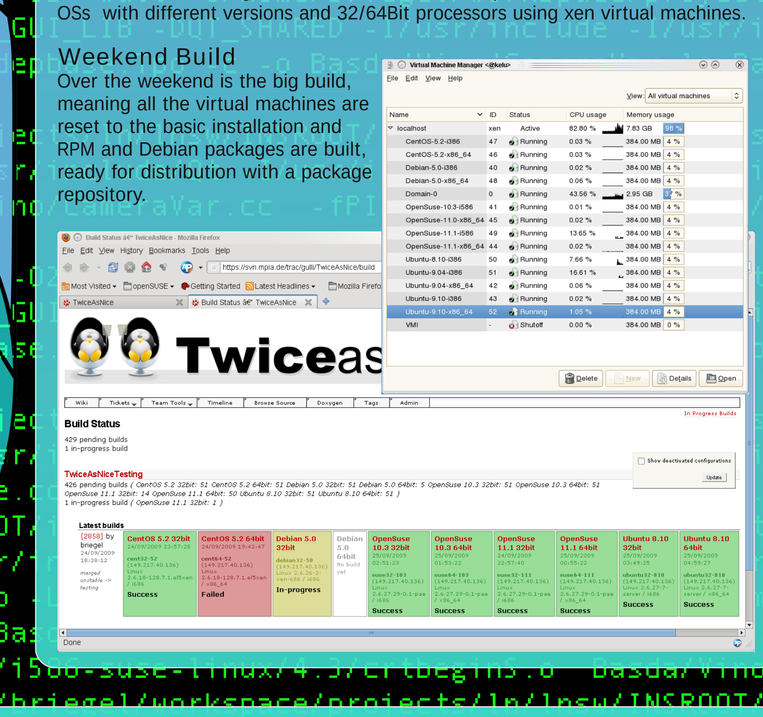
Before releasing a new version of the software several steps have been introduced to improve release quality.

Nightly Build

Every night from Monday to Thursday is the software merged from the unstable branch into the testing branch and afterwards compiled and tested on various OSs with different versions and 32/64Bit processors using xen virtual machines.

Weekend Build

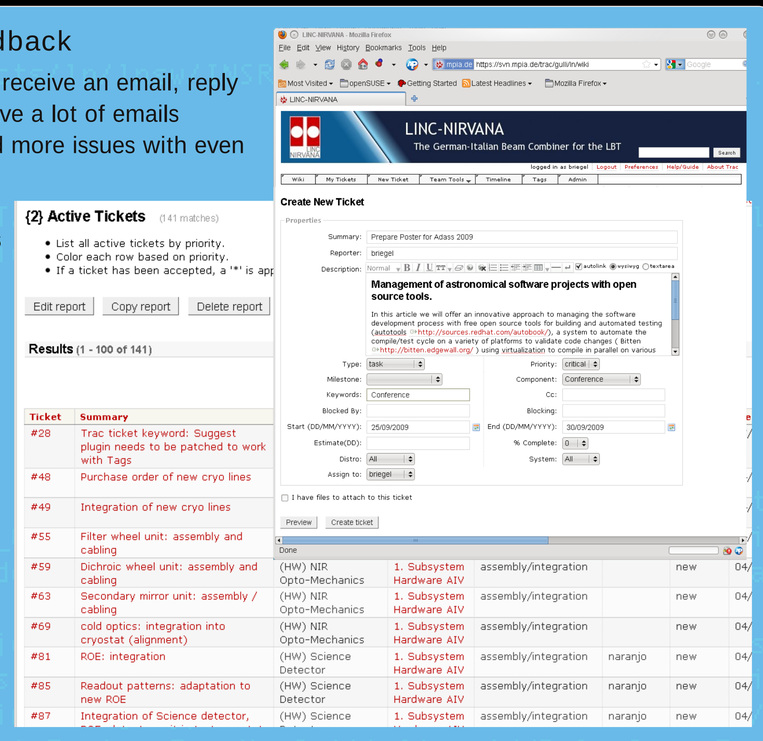
Over the weekend is the big build, meaning all the virtual machines are reset to the basic installation and RPM and Debian packages are built, ready for distribution with a package repository.



5. Write me a Ticket - User feedback

The traditional way of user feedback is to receive an email, reply to it, and reply again. At one point you have a lot of emails spreading to various people and more and more issues with even more emails. Or even worse user feed back by oral agreement. This leeds to misunderstandings and most of the issues will be forgotten anyway after awhile. Best practice whould be, that users write tickets:

- All the issues are in a central place.
- Users are notified about the status of the issue.
- Screenshots or other attachments can be added to the ticket.
- Developers and Users can reply to tickets.
- All the tickets can be formated same as wiki pages.
- Tickets cannot be deleted, only closed.



Acknowledgement

We thank all our colleagues from the LINC-NIRVANA Team and all other projects for there feedback using the software, our system group for setting up the web server and especially the Trac team and the Trac Hackers for there wonderful peace of software released as Open Source, freely available to everyone - long live the freedom of choice!

References

[1] Autotools - http://sources.redhat.com/autobook/
[2] Xen - http://www.xen.org/
[3] Subversion - http://subversion.tigris.org/
[4] Trac - http://trac.edgewall.org/
[5] Trac Plugins - http://trac-hacks.org/
[6] Bitten - http://bitten.edgewall.org/