Software development infrastructure for the FAIR experiments

F. Uhlig
GSI, Darmstadt, Germany

Beside the framework itself the FairRoot core team also provides software development tools for the experimental groups using FairRoot as base for their own developments. This tools should help the developers to concentrate on his main topic, the development of the experiment specific simulation and analysis software, which includes also the independence of a specific code development tool. Each user can work with the tool he is used to. The tools used by the developers ranges from a simple editor like *vi* to IDE (Integrated Development Environments) like *XCode*. The CMake [1,2] based build system will created the native build files for each environment out of simple configuration files which are stored in a portable text format.

The main components of the software development infrastructure are the source code repository as well as the test and build system. The so called software testing server is used to store and view the information produced by instances of the test and build system. The project management software provides a web interface to the source code repository, a bug tracking, and a wiki system which are tightly integrated. This makes it for example possible to link bug tickets directly to the problem in the code, which allows the user to move from the description of a problem to the erroneous code block with one click inside the web browser.

Some components will be described in more detail below. A detailed description of most of the components can be found at [3].

Source Code Revision Control System

As revision control system we use Subversion (SVN) [4] which is the de facto standard for central revision control systems. The main SVN repository at GSI hosts beside the FairRoot framework also the source code of all supported experiments. This somewhat complicated repository structure has some advantages. First of all it is easily possible to couple the experiment specific code with the base framework without having the need to copy the code of the FairRoot framework. The base framework is included to the experiment specific code (CbmRoot, PandaRoot etc.) as external packages. This setup is completely transparent for the user so he will not see that some of the code comes from a different repository. Another benefit of this SVN setup for the experiments is that we can provide features like continuous integration and backup of the source code on every commit to the repository. This support is triggered by the repository itself.

Project Management

We use *trac* [5] as project management software which combines the advantages of a wiki and a bug tracking system with a powerful web front end to our subversion repository. Any of these tools alone is very useful and may be even needed to manage a larger software project but the tight connection of these tools under one hood even enhances the value of the single tools. All tools use a wiki markup language that allows to easily create hyperlinks. This simplifies the connection from a bug report to a line of code in a specific revision of the software or make it possible to create a hyperlink from the commit message to a closed feature request.

The source code browser allows to navigate in the source tree, check older revisions or commit messages. This includes also as diff-like comparison between different code revisions.

The issue *tracking system* is absolutely necessary to keep track of the development process. This includes also the ability to report software problems or request new features in a well defined and traceable way. I also makes sure that the request is assigned to the correct developer.

trac also allows to define milestones which define states of the project like a stable version or the implementation of a feature. The tickets can be assigned to such a milestone and the roadmap shows the milestones together with the open and closed tickets, which shows by this means the progress towards the milestone. The timeline shows all events like new commits to the repository or new tickets in the project ordered in time.

The wiki allows to create and organize a collection of linked information without great affords. Due to the easy creation of cross links the complete system becomes a valuable source of information. If there are still features missing they can be added using a plugin mechanism. There are already many plugins like support for other version control systems or a spam filter available.

References

[1] K. Martin and B. Hoffmann, *Mastering CMake. A Cross-Platform Build System*, Kitware Inc., 2007

[2] http://www.cmake.org

[3] F. Uhlig and M. Al-Turany, PoS(ACAT2010)043

[4] http://trac.edgewall.org/

[5] http://subversion.apache.org/