# Quattor Software Process

*C. Loomis (CNRS/LAL)*

*Quattor Workshop (London)*
*11-13 March 2009*



Information Society and Media

# Current Situation

- Quattor build tools (mostly) standardized.
- Some ad hoc checks of conventions.
- Nightly build performed from trunk.

- No “dashboard” of current state of code base.
- Few automated tests of quattor tools.
- Little gathering of documentation.
- No quality checks of the (perl) code.

- Could do much to improve the quality of the code base, make documentation accessible, and improve maintainability.

# Continuous Build

- **Suggest moving to a system like Hudson for continuous builds of the code base.**
  - https://hudson.dev.java.net/
  - Will build when there are changes to the code base.
  - Can define multiple “builds” and hierarchies of “builds”.
  - Sends notifications when build fails (and when it works again).
  - Collects output and test results for easy review.
  - Provides “dashboard” of current and past results.

# Coding Standards

- **Coding standards and development guidelines:**
  - A couple good documents exist because of Luis.
  - Not automatically checked and largely ignored.

- **Should start running something like Perl::Critic systematically over the code and providing output to developers.**
  - http://search.cpan.org/~elliotjs/Perl-Critic-1.096/
  - http://perlcritic.com/
  - Uses guidelines from “Perl Best Practices” as a base.
  - Highly configurable with large set of code checks.
  - Can be extended to include our own requirements.

# Unit Tests

- **Having a good set of unit tests makes refactoring and changing code more reliable.**
  - Many choices for framework; perl's own may be easiest.
  - Especially important when applying standards to code.
  - Need to implement in a way that many tests come along for "free", especially for things like the configuration components.
  - More important need to adopt a "test-driven development" mentality so that important information about behavior is documented.

# Extracting Documentation

- **Documentation:**
  - Not centralized.
  - Some (like component documentation) is difficult to find.
  - Automated extraction of documentation may help with this.

- **Something like Perl::Tidy may help with this.**
  - http://perltidy.sourceforge.net/
  - Extracts POD documentation formats (optionally) as HTML.
  - Generated documentation can contain tidied perl code.
  - Could be used to reformat perl code automatically.

# Conclusions

- I think that starting to improve the overall software process and tools for quattor will benefit both the users and the developers.

- Would like to see us adopt a plan for moving towards a test-driven development mentality and for adopting automatic checking of the quattor code base.