



The Quattor Release Process

03/07 - Trinity College Dublin

Marco Emilio Poleggi – *CERN-IT/INFN-CNAF*

`Marco.Emilio.Poleggi@cern.ch`



- Goal
- Process overview
 - + Actors
 - + Tools
 - + Procedure
- The *Current Status* page
- Where are we now?
- Open issues

□ Automation

- + In release management: *as high as possible*
- + In release procedure: *99,99%*

□ Ease of management

- + Single configuration point
- + Simple procedures, accessed via unique interface

□ Tight cooperation

- + Balance of freedom vs release operation speed-up
- + Minimal stress on contributors, though prodding a bit ;-)

□ Integration

- + Trackers and documentation systems
- + Autobuild systems

□ Contributors

- + Are responsible of their own modules
 - Coding, CVS, testing, patching, tracking bugs, documenting
 - Receiving feedback from users: they are not a workbench!
 - Keeping up-to-date “official” status information

□ Users

- + Should stay in the loop with no responsibility for testing
 - May confirm issues and solutions, but should not be called to *decide* about them

□ Release manager

- + Is responsible of the release process
 - Procedure proposals and development
 - Calls for phase switch (code freeze, updates, etc.)
 - Release building, packaging, documenting and publishing

□ TWiki

- + Release procedure *documentation*
- + Official *current status page* (see following slides)
- + We need a minimal *developer's convention page!*
 - Coding standards and best practices
 - CVS handling
 - Savannah usage
- + Templates for extra documentation, such as man pages?

□ CVS

- + A *code maturity indicator* should be there, or not?
 - If so, we might need branches again, though something like **foo_stable_x_y_z** should suffice
- + Stricter commit pre-requisites based on **config.mk** tags?
 - Currently only **VERSION** checked for NCM components

□ Make as the **user interface**

- + Targets available for most tasks, from listing changes to build a YUM repository. Unique configuration file **release-list**
 - Integration with ETICS is to come soon
 - Some basic support for TWiki might be useful, such as dumping code snippets for maintainer names

□ Savannah

- + Many improvements made
 - Stricter bug workflow (sorry for the extra load!)
 - Fixed mapping between CVS directories and “Component” field choices
- + Automatic extraction of information still needed, but neither time nor expertise for that (this is the missing 1% for a totally automated release process)
 - For now, contributors could back this up with *their* change-logs (major issues only) sent to the release manager ;-)

Details at [QuattorReleaseProcedure](#) (to be updated, sorry!)

1. *Announce start of release project*

- a) 10-15-days hard deadline for “code (functionality) freeze”. (Do we need a special status, such as **PREPARING?**)
- b) Refresh candidates packages in **release-list**

2. *Beta preparation*

- a) Switch status to **CODE-FREEZE**
- b) Build a *release candidate* (RC) and advertise it
- c) Allow two weeks for testing
 - Rebuild the RC if needed

3. *Release it*

- a) Tag the new release in CVS
- b) Rebuild the *YUM* repository
- c) Switch status to **ISSUED**

- It's meant as a unique official reference
 - + Each developer is responsible of a (small) part
 - Reduce communication overhead during the release process
 - + Primary source for the list of candidate packages composing the next release
 - + Minimal dynamic usage:
 - Switching release status, updating component's status/maintainer
- Some parts could be generated automatically, but manual update still needed
 - + CERN AFS account required, operations not easily “scriptable”
 - + Links to standard HTML pages, periodically refreshed, could be used, as for NCM components' man pages
 - + CVS should be the unique source
 - Module/component's status (special tags needed)
 - Maintainer name

□ Possible add-ons

- + Links to release notes (what you would like to see in the release ChangeLog)
 - Dedicated files in CVS?
- + Links to Savannah items
- + Links to CVS views
- + ...any other wish? Maybe I'm asking TWiki too much ;-)

□ But...

- + Links clog the TWiki code: scripting and forms could help, though it's a bit tricky
 - Better off embedding some HTML: any expert/volunteer here?

- 99%-automated procedure
 - + Release ChangeLog still made by hand
- Parallel build for different platforms
 - + Via a CVS checkout shared on AFS
- Delivery time: 2 hours (instead of 2 days)
 - + Documentation overhead not included!
- Automatic conversion of templates
 - + It could be used to generate different *templates distributions*
- Lean enough to be handled by one person
 - + Still something to drop, such as the *release plan*
 - Long-term tasks in Savannah as standard bugs

- Orphaned modules and components
 - + Who wants to adopt them?
 - Missing or untraceable maintainer = unmaintained component
 - + I'd like to get rid of **MAINTAINER** files in favor of tags in **config.mk**
- ETICS integration
 - + Project area is ready
 - + Still to be decided how to enhance the build framework
- Test framework
 - + Still minimal support
 - All core modules should have unit tests
 - Components should have at least a template syntax check
 - + “Relocatability” problem!
- Release updates could be managed as full releases

 **quattor**

<http://quattor.org/>