Sardana Collaborative Work

1 Collaborative work	**2** Releases
3 Integration of contributions	**4** Code review & CI

Collaborative work

- Examples:
  - SEP6 development: DESY (Teresa) & ALBA
  - Custom recorders in Sardana: DESY (Jan) & ALBA

Collaborative work

- Examples:
  - SEP6 development: DESY (Teresa) & ALBA
  - Custom recorders in Sardana: DESY (Jan) & ALBA
- Experience:
  - Driving discussions on the devel mailing list:
    - Allows others to jump in with valuable feedback.
    - Discussion archives are available for the future use.
  - Call vs. Mail – Call definitely wins :)
  - Vide calls with sharing screen allows analysis of the code, documents

Collaborative work

- Examples:
  - SEP6 development: DESY (Teresa) & ALBA
  - Custom recorders in Sardana: DESY (Jan) & ALBA
- Experience:
  - Driving discussions on the devel mailing list:
    - Allows others to jump in with valuable feedback.
    - Discussion archives are available for the future use.
  - Call vs. Mail – Call definitely wins :)
  - Vide calls with sharing screen allows analysis of the code, documents
- Develop in pairs between institutes seems to be the way to go, what is your opinion

Collaboration in development of macros & controllers (m&c)

- Do you reuse the m&c of others?
- Do you consider easy to browse the repositories?
- Do the m&c provide enough documentation to use them?
- Controllers and macros dependent on other software:

  Where to place this software if it is not public yet?

Questions? & Comments?

Sardana & Taurus Releases

- Now: 2 releases (PyPI) per year.
- Release process is documented with details in the wiki.
- The release process got simplified & unified thanks to :
  - use of gitflow
  - use of semantic versioning
  - migration of the documentation to ReadTheDocs
- Now any of the institutes could do the release.
- To be improved: guides of the manual tests.
- Who uses master, develop or institute-specific versions?
- Are 2 releases per year enough?

Questions? & Comments?

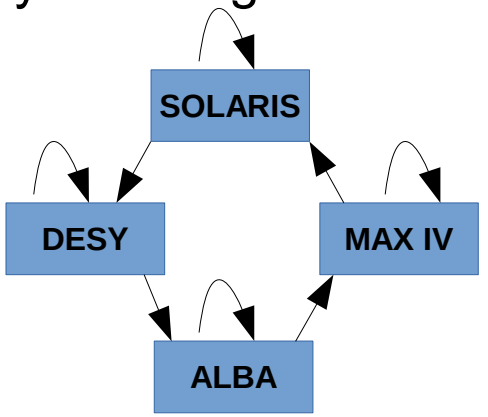
Integration of the contributions

- Current problems with the integration tools/workflow:
  - Long queues of the patches/branches to integrate.
  - Unbalanced load on integrators.
  - No centralized register of the patches/branches pending to be integrated.
  - No relation of the ticket and patch (apart from the e-mail subject).
  - Current workflow require tests to done by the integrator locally.

Integration of the contributions

- Current problems with the integration tools/workflow:
  - Long queues of the patches/branches to integrate.
  - Unbalanced load on integrators.
  - No centralized register of the patches/branches pending to be integrated.
  - No relation of the ticket and patch (apart from the e-mail subject).
  - Current workflow require tests to done by the integrator locally.
- How to improve it?
  - Chain/circle rule of integrations
  - SEP4, TEP3, TEP14

    (who? & when?)



Questions? & Comments?

Code review & CI

- Setup online code review tools (gerrit?, phabricator?, ...) & continuous integration tools (jenkins?, buildbot?, ...)
- Interesting training about: git, gerrit, jenkins and redmine from FRM-II.

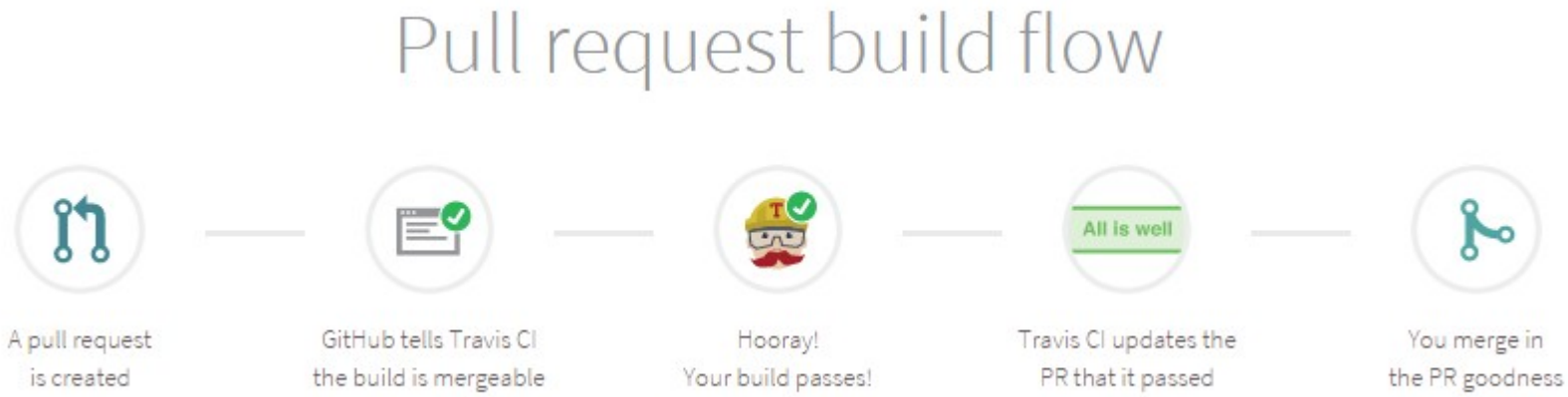
Code review & CI

- Setup online code review tools (gerrit?, phabricator?, ...) & continuous integration tools (jenkins?, buildbot?, ...)
- Interesting training about: git, gerrit, jenkins and redmine from FRM-II.
- Or move to github and travis?
  - What is your experience with Github?
  - We would need to created Github accounts.
  - Current workflow is based mainly on emails and Github is web based, is it a problem?
  - Migration of tickets, wikis, repos from SF to Github?
  - Tango project is on SF.

Pull request on Github

1) Developers work in topic branches on their forks.
2) When the contribution is ready they pull request using the Github webpage (commit history and changes can be previewed prior to the submit).
3) Pull request notifies the rest of the developers via email.
4) Github checks if the branch can be successfully merged.
5) Discussion starts allowing: questions, comments, file and line notes.
6) Further commits pushed to the branch will automatically be added to the pull request.
7) Pull request gets reviewed and finally merged to the base branch.
8) The whole process history is maintained on Github.
9) Pull request dashboard allows easy filtering and sorting.

CI on Travis



- Pull request build flow seems to fits to our needs.
- First filter of contributions for the integration manager.
- Travis recently included Docker in its services:
  - pull Docker images & run Docker containers
  - build Docker images & push them to the registry

Questions? & Comments?