SVN to Git Transition Planning (Version 0)

Ville Suoranta (ed. Mark Rawlings)

General Issues

Our current branching model has ASAP maintained as a separate repository, and the future model would have ASAP and Casacore as their own individual entities. These have historically been brought together in a separate branch for releases. This means that the branches and trunk have different models and must somehow be reconstructed. Future branches in the sources will not come over cleanly (due to metadata, commit numbers/hashes), and hence we would still need to perform the release branching in SVN using the old model.

Strategy options

1) Centralized

Pull all of the sources to a central NRAO managed repository. The conversion is done once and development work and builds continue from the new repository.

Pros:

Fairly simple conversion process;
Full control of the source tree in a single location;
Probably easier to make work with an "off the shelf"-type CI approach.

Cons:

Some of the existing stakeholders may prefer to retain their current level of control of existing non-NRAO repositories (maybe mitigate this somewhat by making them gatekeepers?);
Everything has to be in place on the date of conversion. This would include developer training and build system updates.

2) Hybrid

Maintain SVN repositories in parallel with NRAO Git repository and the Casacore Github. Developers would commit their work to SVN repositories and GitHub, and the changes are then pulled to the NRAO Git repository.

Pros:

1. In theory, the transition is easy and gradual and allows developers to keep working on SVN until the build system changes are complete.

Cons:

1. Requires managing two different version control systems in parallel.
2. Due to the way that the trunk and branches are set up in SVN, both would need to be translated separately to Git.
3. Little control over GitHub repository. This issue might be mitigated somewhat by judicious assignment of gatekeeper roles?

Variants:

1.1) Convert all of the source systems to one Git repository and update periodically.

Pros:

Easy to tag/branch. Easy to cut off from source systems and re-organize.

Cons:

1) Currently only a 2-stage approach works due to having multiple source repositories for trunk and a single repository for branches.
2) Requires good control of the Casacore repository.

1.1.1) Point directly to the sources and create subtrees

Pros:

1) Slick and straightforward

Cons:

1) Fetching/merging changes does not work very well.
2) Branches are not included.
3) Rebasing from SVN does not really work or is difficult.
3) Based on the above, effectively ends up really being an all-at-once solution.

1.1.2) Multi-tiered conversion and sync

SVN repositories are pulled to git repositories. Branches are treated as complete entities, but trunk is stitched together from multiple sources and pulled together as sub-trees of a single repository (see pdf attachment for a diagram).

Pros:

Updates work for trunk and branches.

Cons:

Complex model that requires multi-step updates

1.2) All source systems to separate Git repositories

Pros:

Each sub-system can be managed separately.

Cons:

Each sub-system has to be managed separately.
Shifts some of the issues to the build system.