

SoundSoftware Code Repository

Some provisional decisions and open questions
Chris Cannam, May 2010

Provisional decisions

Primary version control system for code: Mercurial

Code forge application: Redmine

Server configuration: Ruby on Rails (RailsEE),
Phusion Passenger, Apache 2

Mercurial?

Distributed system:

Better than centralised for maintaining software that we do not claim to “own”

Better reassurances for backup etc (each checkout is a complete repository)

Easier to get to grips with than git, especially for non-geeks

Portable, fast, friendly, well documented

See e.g. hginit.com (tutorial), hgtip.com

Redmine?

Easy to deploy, pleasant to use

Good support for Mercurial repositories

Integrated wikis, issue trackers, forums

Flexible theming and customisation

Though you do need to know Ruby to code any serious extensions

Memory hungry, but pretty quick with Passenger and >1GB RAM on server

Open questions

Authentication?

Roles: Who gets to do what?

Versioned data repository?

Build integration?

Backups?

Authentication?

- Redmine (and Drupal, now on website) have separate user databases but can be extended to authenticate externally
- For QM users, authentication against CS department LDAP server would be ideal
 - Is there a “partner” level – authentication against someone else's LDAP?
- External users need to be able to register themselves (with admin approval)

Roles?

- Need to formalise:
 - public project
 - private project (“ours” or someone else's)
 - project we are only tracking
- Control all aspects of a project:
 - Use of Redmine
 - Direct access to Mercurial repository via HTTP
 - How to restrict both read and commit access to (specific subsets of) LDAP-authenticated users?

Versioned data repository?

- Many projects have ad-hoc test data
- This doesn't go well in a distributed SCM
 - Using big files hurts when you're copying the entire repository with every checkout
 - Every update to a binary file adds the whole file size to the working tree – no deltas
- Do we want a Subversion repo as well?
- Is this different from our “proper” experimental data repository?

Backups?

- Major element of reassurance for users here

Build integration?

- Still a completely open question; build server management software hasn't been evaluated yet
- How to avoid this being a complete mess of brown paper and string?