Subversion Repository Structure.

Martin Killenberg

12th June 2015

MSK Collaboration Workshow 2015, Warsaw, Poland

Repository Contents

What has to go into the repository?

Everything that is written manually

- Source code (header and programme code, scripts, Makefiles)
- Doxyfiles, manually written documentation sources
- Manually written VHDL code
- Design documents for PCBs

What does NOT belong into the repository?

Everything that is (automatically) generated

- Executables, libraries, object code ($\rightarrow$ Debian package server)
- Temporary files
- Generated documentation (Doxygen output) ($\rightarrow$ Jenkins)
- Bit files, generated map files ($\rightarrow$ Jenkins)
- Measurement results

SVN Repository Structure

trunk

- main development
- SW: should be stable (compiles and tests are OK)

branches

- features which need more than one commit
- possibly unstable
- advanced users, empty in most projects

tags

- stable, released version
- usually copies/snapshots of the trunk
- SW: versions are packaged and deployed
- HW: versions which are actually build

Production Systems

- Only use tagged versions!
- No SW/FW without tag in the tunnel!

Repository Organisation

Everything with its own release cycle needs trunk/tags/branches.

Container repositories with sub folders, each with trunk/tags/branches

- Useful for related projects (allows cross merging)
- Only one access list to maintain

Many small repositories, only one trunk/tags/branches in the repository root
Recommended by the SVN admins in Zeuthen

- Individual access list for each repository
- Can be grouped on the Zeuthen svn server

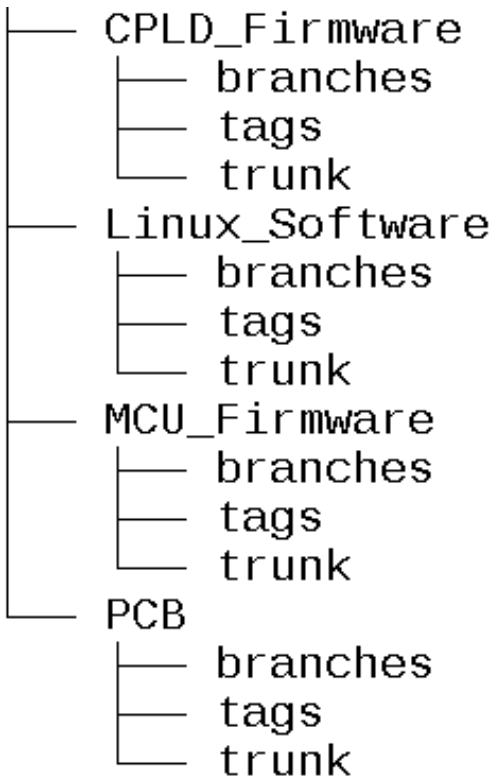
Wrong: ~~**Container repositories with only one trunk/tags/branches**~~,
and many projects inside of trunk

- You cannot do reasonable tags (and branching)
- The repository becomes a data dump

Example: Fuse Relay Board (FRED)

Components:

- Hardware (board design)
- Firmware
- C++ library
- GUI



Hardware (PCB)

- Different revisions of the board are tagged

Firmware

- Two chips (MCU and CPLD)
- Each with its own release cycle

Linux Software

- Library and GUI are closely coupled
- Always released together (same release cycle)

Discussion: Where to put which project?

MSK Firmware SVN server

- Behind the firewall
- Only for MSK/DESY members

Zeuthen SVN server

- Maintained by (Zeuthen) IT
- Publicly accessible, also for check-in
- External users via email
- Developers managed by repository admin

MCS4 GIT server

- Publicly readable
- Developer access only for DESY members
- New DOOCS servers go here!

DESY Stash GIT server

- Maintained by IT
- Publicly accessible, also for check-in
- External users possible?
- No experience yet

~~**MCS4 CVS server**~~

- Outdated, do not use any more!