
Subject: GIT essentials

Posted by [kohait00](#) on Mon, 11 Jan 2010 11:00:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

hi people,

i have been using svn in my company now for some time and it was ok, sometimes it was a pain. and one time we even had a crash on the server, boom, the repo was gone, the history too. my fault, though. should have done backup before. well, life continued with the revision that was on my work laptop. with git, that cant happen.

the trunk/branches/tags model in svn (which is not obligatory though, but used widely) is kinda strict in a sense. it does not motivate a developer much to try and experiment with the codebase, because one fears the whole commit/revert/reread and filter the changes stuff.. and you always need the repo connection for looking up the history, the graph etc.. with git that is far esier.

admittable, git has a hard learning curve in the beginning, since the concepts are remarkably different from svn, so svn users espacially have their what?s and how?s at the beginning. but one could simply present them some tutorials and explain the basic things, so they get the picture.

the fact that everyone has its own repository, and mirek would need to pull is both an advantage and a disadvantage. he would not need to care much about commit right of other users, because *HE* needs to decide from whom to pull or not, this is all. he does not need to give anyone the commit right to its own git repo at all, basicly. so user management is franly different. mirek could be able try out some ideas and reject them without comment or polluting its own repo master branch. having a bunch of remote branches from *trusted* co contributors would make it easy to keep track. but probably this would make more work for him (more decision work especially), whereas now, he grants commit right to some users, trusting them not to pollute the trunk and does not need to take care of it anymore (but this is also possible in git). but if we consider that linus has as little time as mirek and manages to handle kernel development with git (it actually was developped for that purpose), then it should be even easier.

i'd personally prefer to switch from svn to git, it is way superior to svn and many other scms (as far as i understood) and it should even be preferred over mercurial and other commercial projects, cause git will have a much wider accaptance and be wider used in future due to its open source nature. one can even say its more *standard* now. (look at all the major projects one by one dropping svn and heading for git).

we clearly have to destinct 2 things concerning usage of git for u++:

- 1) usage of git as scm for the Ultimate++ project *itself*
- 2) providing a git interface (among others, just like svn) in TheIDE to sync the user's project codebase when he uses git for it (has nothing to do with the ultimate++ project itself at all)

i think that was not clear cut apart in the preceeding posts

beeing using svn for quite some time, i started to switching to git, and still learning, have strived some of the tutrials and infos on the net over git. and so far, became more and more fixed in my

decision to stay with git.

hope U++ will consider to move to git too. sorry for the long post

PS: Linus, the godfather of GIT making a really enjoyable talk about GIT itself, at google talk, focusing and adresssing most of the topics we descuss here. is worth watching, makes some harsh jokes . this video made me think about switching. is quite long, but woth it.
<http://www.youtube.com/watch?v=4XpnKHJAok8>
