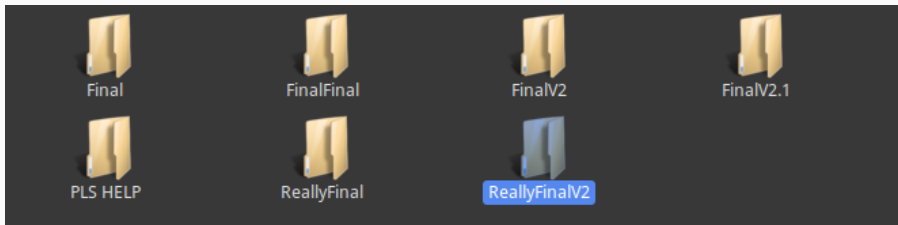


Ist euch so etwas schon einmal passiert?



Das geht auch anders... ohne Dropbox oder Google Docs.

SVN und Git

Versionsverwaltung und Issue Tracking

Simon Labrenz

25.07.19

SVN

Commands

Git

Commands

Tools für Windows

Tools für Linux

Branches, Merges und Rebases

Vergleich Git vs SVN

Issue Tracking

Fazit

SVN

Commands

- checkout
create working copy
- commit
save changes
- add
add file to the svn
- diff
show difference between your work and the svn
repo

Git

Commands

- status
shows unadded, added files and unstaged changes
- checkout
create working copy
- commit
save changes
- add
add file to the local repo

Commands

- pull
get the latest changes from the remote repo
- push
upload local changes to remote repository
- diff
show local uncommitted features

Hosts

- Github
- BitBucket
- GitLab (benötigt eigens gehostete Instanz)
Seit neustem aber auch von FIM
Systembetreuung gehostet.
Anmeldung über *gitlab.fim.uni-passau.de* mit
eurer CIP - Kennung

Windows

- Source Tree
- GitKraken (Github only)
- TortoiseGit
- Fast alle Java IDEs bieten Git Extensions
- Git shell

- Command Line (zusammen mit tig)
- GitKraken (immer noch nur für GitHub)
- Git Cola (klein aber fein)
- Meld (Super für Mergekonflikte)
- IDE PlugIns

Branches, Merges, Rebases

Aus rechtlichen Gründen gibt es anstatt Bildern an dieser Stelle eine kleine Malstunde mit Onkel Simon.

1. Branches
2. Merges
3. Rebases

Weitere Git Features

- Tags / Releases
- Blame
- Bisect
- Issues

Vergleich Git vs SVN

Git vs SVN

1. SVN

- einsteigerfreundlich
- weniger Features
- zentraler Server → aktive Verbindung nötig
- lineare History → einfach lesbar

2. Git

- für Fortgeschrittene
- viele Features (nicht zwingend SEP relevant)
- verteilt → auch offline nutzbar
- Versionshistory mit mehreren Branches → nicht linear

Issue Tracking

Issues

- Bugzilla
- Github / Gitlab
- Gantis
- Redmine (Projekt Verwaltung)

Fazit

Fazit aus eigener Erfahrung

- - - force push deaktivieren
- Git ist in meinen Augen besser, wenn auch mit extra Aufwand verbunden
→ Die Vorzüge überwiegen die Kosten
- Mindestens eine Person sollte sich vorher schon einmal Git beschäftigt haben
- Nichts wird gepusht, was nicht compilet oder getestet wurde → Sonst wird aus git push ein git pfusch!