Git SCM

Crash-Course in Versionsverwaltung and kolaborativem bearbeiten von Textdateien

Dustin Frisch
Fachbereich Angewandte Informatik
Hochschule Fulda
Fulda, Deutschland
E-Mail: dustin.frisch@ai.hs-fulda.de

14. September 2021

Agenda

Was soll das alles?

Warum jetzt genau Git?

Einrichtung und Installation

Aller Anfang ist schwer

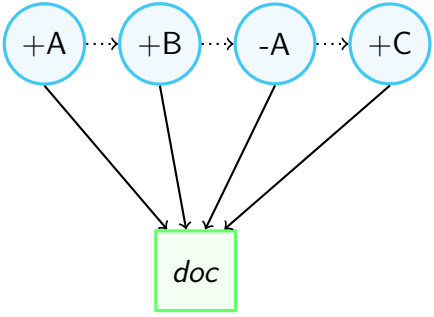
Ein Baum ist ein Baum ist ein Baum

Mein, dein. Das sind doch bürgerliche Kategorien

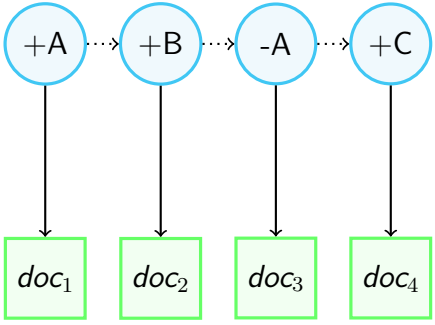
Alles immer bitte gleich richtig machen

Was ist Versionverwaltung?

Ohne Versionverwaltung



Mit Versionverwaltung



Wo kann ich das einsetzen?

- Source-Code
- Konfiguration
- Dokumente
- Daten

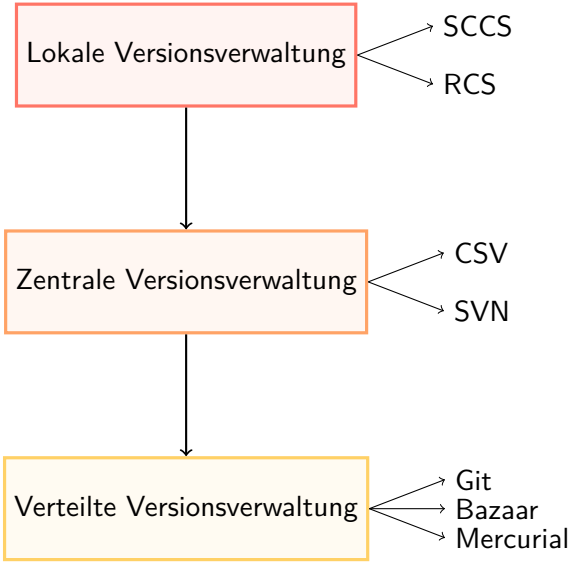
Also alles, was *Text* ist.

Zeitmaschinen

und

Paralleluniversen

Kurze Histore



Git is everywhere

Installation

Windows https://gitforwindows.org/ oder

```
# choco install git
```

Linux `# sudo dnf install git-all` oder

`# sudo apt install git-all` oder ...

macOS `# git --version`

Einrichtung

```
# git config --global user.name "Dustin Frisch"
# git config --global user.email "dustin.frisch@ai.hs-fulda.de"
```

Pause

Neues Projekt

Neuen Ordner anlegen und in diesen wechseln.

```
# git init
Initialized empty Git repository in /home/fooker/tmp/myproject/.git/
```

Den überblick behalten

Eine Datei in dem Ordner anlegen.

```
# git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test.c

nothing added to commit but untracked files present (use "git add"
↪  to track)
```

Der Staging-Bereich

```
# git add test.c

# git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   test.c
```

Mein erstes mal

```
# git commit
main (root-commit) 27cda20] A wonderful commit message
 1 file changed, 6 insertions(+)
 create mode 100644 test.c

# git status
On branch main
nothing to commit, working tree clean
```

Oh, ein wie konnte das nur passieren

Die Datei in dem Ordner wird geändert.

```
# git add test.c

# git commit
```

Was bisher geschah...

```
# git log
commit 7163f25b8e477a43accc28c30f3bd4351a2d72a2
Author: Dustin Frisch <fooker@lab.sh>
Date:   Mon Oct 18 22:12:28 2021 +0200

    Fixed dumb mistake

commit 27cda204ad4caae123b44f1fc8d3ac6645e66d3f
Author: Dustin Frisch <fooker@lab.sh>
Date:   Mon Oct 18 22:07:31 2021 +0200

    A wonderful commit message
```

So langsam wird es

Eine weitere Datei anlegen.

```
# git add example.c

# git commit -m "Added example"
```

Namesgebung ist schwierig

Eine Datei umbenennen.

```
# git status
On branch main
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working
↪  directory)
        deleted:    test.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        main.c

no changes added to commit (use "git add" and/or "git commit -a")
```

Namensgebung ist schwierig

```
# git add test.c main.c

# git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    test.c -> main.c

# git commit -m "Naming fixed"
```

Und weg damit

Eine Datei löschen.

```
# git rm example.c

# git commit -m "This was a mistake"
```

Was hab ich nur getan

Eine Datei ändern.

```
# git diff
```

Keine Fehler

```
# git restore main.c
```

Was werde ich nur getan haben

Eine Datei ändern.

```
# git add main.c

# git diff --staged
```

Nee, doch nicht

Datei nochmal ändern

```
# git restore --staged main.c
```

Alles hat ein Ende

```
# git branch
* main
```

Nur der Baum hat viele

```
# git branch feature-x

# git branch
  feature-x
* main
```

Wer die Wahl hat

```
# git checkout feature-x
Switched to branch 'feature-x'

# git branch
* feature-x
  main
```

Als währe nichts gewesen

Ein paar Commits erzeugen.

```
# git checkout main
Switched to branch 'main'

# git log
...
```

Realität ist subjektiv

Eine paar Commits erzeugen.

```
# git checkout feature-x
Switched to branch 'feature-x'

# git log
...
```

Zusammen was zusammen gehört

```
# git merge main
Merge made by the 'recursive' strategy.
...

# git log
...
```

Pause

So klein und schon bei den Sturmtruppen?

```
# git clone http://git.open-desk.net/git/_.git project
```

```
# git remote -v
origin        http://git.open-desk.net/git/_.git (fetch)
origin        http://git.open-desk.net/git/_.git (push)
```

Jeder fängt mal klein an

```
# git remote add origin http://git.open-desk.net/git/_.git
# git push origin main
...
To http://git.open-desk.net/git/_.git
 * [new branch]      main -> main
```

Zieh, zeih, zieh!

```
# git pull
...
From http://git.open-desk.net/git/_
 * [new branch]      main       -> origin/main
```

Und jetzt alle zusammen

In beiden Ordnern einen Commit erzeugen

```
# git push origin main
...
To http://git.open-desk.net/git/a.git
   ba998dd..a5069cc  main -> main
```

```
# git push origin main
To http://git.open-desk.net/git/_.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to
↪  'http://git.open-desk.net/git/_.git'
```

Ständiges Hin und Her

```
# git pull
...
From http://git.open-desk.net/git/a
 + 07493c7...a5069cc main       -> origin/main  (forced update)
Successfully rebased and updated refs/heads/main.

# git push
...
To http://git.open-desk.net/git/a.git
   a5069cc..baf125b  main -> main
```

Ich meine es nur gut

- Nutzt SSH-Schlüssel mit einem SSH-Agent.
  Tutorials gibt es von GitHub.
- Signiert eure Commits.
  Auch hier hilft GitHub weiter.

Ja, wirklich

- Konfiguriert euch einen Shell-Prompt für Git.
- Macht kleine und strukturierte Commits.

```
# git add -p -i .
```

Falls noch Zeit bleibt

- `# git commit --amend`
- `# git stash`
- `# git blame`
- `# git rebase -i`
- `# git rebase --onto`

Danke

https://ohmygit.org