

1. Git

Im Praktikum wurde das Versionsverwaltungstool *Git* vorgestellt. Beantworten Sie die folgenden Aufgaben:

- (a) Was sind die Vorteile bei der Verwendung von Git im Gegensatz zum Arbeiten ohne Versionsverwaltungstools?

Lösung:

Reddit-User [ikilledkojack](#) fasst es sehr gut zusammen:

Remember the phrase „Too many cooks spoil the broth“?

Imagine you've got a bunch of chefs writing a huge cookbook - thousands of pages at least. Chefs being as they are, there will be some delegation as to who writes what parts, but eventually a few things will happen:

- A chef may want to change the recipe another chef wrote
- A chef may want to find an instruction they wrote earlier, but have now changed to say something else
- All of the chefs eventually will want to combine their notes into the final pages for the book, and not lose track of whose pages are on what page

And this is what version control does - if this analogy doesn't make sense to you, think of it more as a Ctrl + Z on drugs that lets you point the finger.

- (b) Wie unterscheidet sich Git von einem zentralisierten VCS wie zum Beispiel Subversion und erläutern Sie die Vorteile, die sich aus diesen Unterschieden ergeben?

Lösung:

Aus [geeksforgeeks.org](#) (übersetzt):

Anstelle eines einzigen Repositorys, das der Server ist, hat hier jeder einzelne Entwickler oder Kunde seinen eigenen Server und verfügt über eine Kopie der gesamten Historie oder Version des Codes und aller seiner Zweige auf seinem lokalen Server oder Rechner.

Im Grunde genommen kann jeder Client oder Benutzer lokal und getrennt arbeiten, was bequemer ist als eine zentralisierte Quellcodekontrolle, weshalb sie auch als verteilt bezeichnet wird.

-
- (c) Nehmen Sie an, Sie arbeiten an einem Softwareprojekt mit vorhandenen Commits. Sie schreiben Code für eine bereits existierende Klasse und stellen im Nachhinein fest, dass Ihre Änderungen das Problem nicht lösen. Wie lassen sich mit Git Ihre getätigten Änderungen rückgängig machen? Unterscheiden Sie dabei zwischen:

- i. noch nicht geadded

Lösung:

Die Änderungen in einer Datei können z.B. einfach mit einer festen, aber beliebigen Anzahl von Ctrl + Z rückgängig gemacht werden.

Will man aber den Stand einer Datei aus einem bestimmten Branch haben, wählt man:

```
git checkout <branch> <file>
```

- ii. schon geadded

Lösung:

Um git add <file> rückgängig zu machen, wählt man:

```
git reset <file>
```

Will man aber den Stand einer Datei aus einem bestimmten Branch haben, wählt man:

```
git checkout <branch> <file>
```

- iii. schon commitet

Lösung:

Ist der Fehler im letzten Commit passiert *und* man will den gesamten letzten Commit rückgängig machen, wählt man:

```
git reset [--soft | --hard] HEAD~1
```

--soft lässt die Änderungen an den Daten in der lokalen Kopie erhalten, wohingegen --hard auch diese Änderungen löscht und damit potentiell Arbeit verloren geht.