

## AUFGABE 1: QUELLCODEVERWALTUNG MIT GIT

Ziel dieser Aufgabe ist es, ein Gefühl für die dezentrale Versionsverwaltung mit `git` zu bekommen. In dieser Aufgabe erhalten Sie von uns ein bereits mit einer Versionshistorie befülltes `git`-Repository. An dieser Versionsgeschichte werden Sie eine Reihe von Änderungen vornehmen.

Nach Bearbeitung dieser Übungsaufgabe sollten Sie verstanden haben wie man Inhalte zu einem `git`-Repository hinzufügt und entfernt, wie man mit einem entfernten Repository kommuniziert, wie man Geschichte umschreibt, und vor allem was man machen muss, wenn mal etwas schiefgeht.

Notieren Sie sich für die Abgabe, welche Befehle Sie in den einzelnen Teilaufgaben verwendet haben und was Sie durch die Verwendung des Befehls erreichen wollten. Dokumentieren Sie bei jedem Schritt die Ausgabe von

```
git log -n4 --graph --format=oneline
```

### *Aufgabenstellung*

1. *Klonen*: Verwenden Sie `git` um die Übungsvorgabe herunterzuladen:

```
git clone https://www4.cs.fau.de/Lehre/SS14/V_VEZS/Vorgaben/astime.git
```

2. *Log*: Verschaffen Sie sich nun mittels `git log` einen Überblick über die Versionsgeschichte. Wenn Sie wollen, können Sie hierfür auch ein graphisches Werkzeug wie `git cola` oder `gitk` verwenden.

3. *Hinzufügen*: Fügen Sie ihren Namen und Ihre E-Mail-Adresse zur Datei `README` zum Abschnitt „Credits“ hinzu. Legen Sie anschließend einen Commit mit dieser Änderung an. *Wie sieht die Versionsgeschichte nun aus?*

4. *Rückgängig machen*: Verwenden Sie nun `git reset --hard HEAD^` um den letzten Commit rückgängig zu machen. Betrachten Sie die Versionsgeschichte. *Sehen Sie einen Unterschied zwischen dem Ausgangsrepository und dem aktuellen Zustand?*

5. *Zeitreise*: Betrachten Sie nun die Ausgabe von `git reflog`. *Was sehen Sie hier?* Suchen Sie im Reflog nach dem Zustand, in dem sich

das Repository am Ende der Teilaufgabe „Hinzufügen“ befand und stellen Sie diesen mit Hilfe von `git reset --hard` wieder her!

6. *Zusammenarbeit*: Ändern Sie in der Datei `modern.astimerc` den Wert des „handThickness“-Eintrags von 70 auf 50 und legen Sie einen Commit für diese Änderung an. Ziehen Sie nun aus dem Repository [https://www4.cs.fau.de/Lehre/SS14/V\\_VEZS/Vorgaben/astimePatched.git](https://www4.cs.fau.de/Lehre/SS14/V_VEZS/Vorgaben/astimePatched.git) dessen Commits mit Hilfe von `git pull` nach. Beheben Sie den dabei entstehenden Konflikt – Sie können `git status` verwenden um in Erfahrung zu bringen, welche Dateien von dem Konflikt betroffen sind. Achten Sie dabei auf den Erhalt Ihrer Änderung. Betrachten Sie nun die Historie – *wie sieht diese aus?*

Verwenden Sie jetzt `git rebase` um den Merge-Commit zu beseitigen und sehen Sie sich die Historie an.

7. *Geschichte neu schreiben*: Führen Sie einen interaktiven Rebase durch, in dem Sie die beiden Commits, die den Konflikt betreffen miteinander verschmelzen.

8. *Umsortieren und verschmelzen*: Führen Sie einen interaktiven Rebase auf dem Commit vom Montag, den 27. Mai 2013 um 18:45:34 Uhr durch und gruppieren Sie diejenigen Commits, die Warnungen des Übersetzers betreffen. Verschmelzen Sie diese Commits anschließend miteinander, so dass nur noch ein Commit übrig bleibt, der Übersetzerwarnungen in Ordnung bringt.

*Notieren Sie sich jetzt die Ausgabe von `git reflog` für die Abgabe.*

9. *Hochladen*: Passen Sie die Konfigurationsdatei des Repositories so an, dass das Remote `origin` auf Ihr Gruppenrepository und das Remote `vorgabe` auf unser Vorgaberepository verweist (siehe Übungsfolien). Laden Sie nun Ihre Versionsgeschichte in Ihr Gruppenrepository hoch.

10. *Abschließende Frage*: *Angenommen das zweite Repository, aus dem Sie gezogen haben wäre öffentlich zugänglich – wieso wären die Rebase-Schritte, die Sie durchgeführt haben dann eine schlechte Idee?*

## Hinweise

- Erforderliche Dateien: keine Vorgabe
- Bearbeitung: Gruppenarbeit
- Abgabezeit: 08.05.2014
- Fragen bitte an [i4ezs@lists.cs.fau.de](mailto:i4ezs@lists.cs.fau.de)