



Workshop: Git und Social Coding (Blatt 02)

Diese Aufgaben sollen Sie darauf vorbereiten in Ihrer Projektgruppe (mit ca. 6 Teilnehmern) gemeinsam zu programmieren.

Bearbeiten Sie auch in diesem Workshop die Aufgaben in Ihrer Projektgruppe. Das heißt, sie arbeiten erneut in Kleingruppen von ca. 2 Personen, nutzen aber mit Ihrem Team ein gemeinsames Github-Repository.

Es ist nicht weiter schlimm, wenn Sie mit den Übungen in der Zeit nicht fertig werden. Das Wichtigste ist, dass Sie ein grundlegendes Verständnis entwickeln, auf dessen Basis Sie dann nochmal in Ruhe zu Hause aufbauen können.

Viel Erfolg!

Aufgabe 1: Umgebung einrichten (ca. 5 min)

Sie wollen gemeinsam mit Ihrem Team an einem Softwareprojekt arbeiten. Bevor alle mit dem Programmieren anfangen können, muss zunächst ein gemeinsames Remote-Repository auf github angelegt werden:

1. Bestimmen Sie eine Person aus dem Team, welche ein Übungsrepository auf <https://github.com/ps-tuebingen-tsp-2016> anlegt.
2. Geben Sie über "Settings > Collaborators & teams > Teams > Add a team" Ihrem Team Lese- und Schreibrechte auf dem Repository.

Fahren Sie nun in Kleingruppen fort.

3. Klonen Sie das Repository, um an einer Kopie auf dem eigenen Rechner arbeiten zu können.
4. Überprüfen Sie mit `git remote -v`, welche Remote-Repositories mit Ihrem lokalen Repository verknüpft sind.

Lernziele

Nach der Bearbeitung dieses Arbeitsblattes haben Sie gelernt

- ✓ Remote Repositories zu klonen
- ✓ lokale Änderungen mit dem Remote Repository abzugleichen
- ✓ entfernte Änderungen mit dem lokalen Repository zu synchronisieren
- ✓ Pull Requests auf Github zu erstellen, kommentieren und zu mergen.

Wichtige git-Befehle für diese Übung

```
$ git clone [URL]
```

Richtet eine lokale Kopie eines Remote-Repositories ein.

```
$ git remote -v
```

Listet die Remote-Repositories auf, die mit dem lokalen Repository verknüpft sind.

Aufgabe 2: Gemeinsam an Code arbeiten (ca. 15 min)

Bisher ist das Repository leer -- Zeit etwas Code zu schreiben. Arbeiten Sie nun "isoliert" in den Kleingruppen, d.h. jede Kleingruppe bearbeitet die Aufgaben alleine, ohne mit den anderen über den Code oder das gemeinsame Repository zu sprechen. Sie können sich weiterhin austauschen, indem Sie die Kommunikationsmittel verwenden, die Ihnen Github zur Verfügung stellt.

1. Erstellen Sie eine neue Datei in Ihrem Arbeitsverzeichnis (im Branch `master`) und committen Sie diese.
2. Teilen Sie Ihre Änderung mit den Anderen, indem Sie diese "pushen". (Dies wird zunächst fehlschlagen, wenn Ihre Kollegen schneller waren.)
3. Ziehen Sie sich die neusten Änderungen mit `git fetch`. Sollten Sie noch lokale Commits haben (`git branch -vv` zeigt "ahead" an), dann mergen Sie die Änderungen Ihrer Kollegen in Ihren `master` branch.
4. Wiederholen Sie Teilaufgabe 2 und 3, bis jede Teilgruppe mindestens eine Änderung beigetragen hat.

Tip: Ein einfacher Weg ständige Passwordeingabe zu vermeiden ist folgende Konfiguration:

```
git config --global credential.helper cache
```

Wichtige git-Befehle für diese Übung

```
$ git push -u [remote-name] [branch-name]
```

Aktualisiert den Branch `branch-name` im Remote-Repository `remote-name`. Die Option `-u` richtet außerdem `remote-name/branch-name` als upstream Branch für `branch-name` ein.

```
$ git fetch [remote-name]
```

Aktualisiert die lokale Kopie von `remote-name`.

```
$ git branch -vv
```

Die Option `-vv` zeigt zusätzliche Tracking-Information zu den Branches an.

Bitte geben Sie Bescheid, wenn Ihre ganze Gruppe mit dieser Aufgabe fertig ist.

Aufgabe 3: Code Reviews mit Pull Requests (ca. 20 min)

Nach einiger Zeit ist Ihnen im Team aufgefallen, dass manchmal Code committed wird, der gar nicht funktioniert. Dies stört alle Anderen in ihrer Arbeit. Deshalb einigen Sie sich darauf, Code nur noch in `master` zu mergen, wenn dieser mindestens von einer anderen Person kontrolliert wurde.

1. Erstellen Sie einen neuen lokalen Branch
2. Arbeiten Sie auf diesem Branch (d.h. machen Sie mindestens eine Änderung, die sie committen)
3. Veröffentlichen Sie den Branch im Remote Repository
4. Verwenden Sie die github Web-Oberfläche, um einen Pull Request zu erstellen.
5. Laden Sie einen Teamkollegen, durch Erwähnung seines Benutzernamens ein, Feedback zu dem Pull Request zu geben.
6. Wenn ihr Kollege einverstanden ist: Mergen sie den Branch und löschen Sie ihn danach.
7. Wenn ihr Kollege nicht einverstanden ist: Machen sie eine Änderung und wiederholen Sie ab Teilaufgabe 5.