Git Übung

Diese Übung besteht aus drei wesentliche Teilen:

1. Die Arbeit mit einem lokalen git Repository.
2. Die Zusammenarbeit zwischen zwei repositories.
3. Die Zusammenarbeit über ein zentrales git Repository.

Es werden jeweils elementare git Befehle verwendet. Es wird komplett lokal am eigenen Rechner gearbeitet. Auch der Austausch zwischen verschiedenen Repositories erfolgt ausschließlich lokal.

GIT Übung – Arbeit mit dem lokalen Repository

1. **Installieren Sie GIT auf Ihrem Rechner**
   ➔ *unter https://www.git-scm.com/downloads downloaden.*
2. **Legen Sie ein kleines Projekt in einem geeigneten Pfad an. Z.B. unter** *D:\projekte\uebungsprojekt*
3. **Legen Sie ein paar neue Dateien unter dem Projektpfad an.**
4. **Legen Sie ein Git repository an.**
   1. Im Projektverzeichnis eine cmd Box öffnen.
   2. > git init
      ➔Meldung *„initialized empty repository in <Pfad>/.git“*
5. **In Ihrem Projektverzeichnis ist jetzt ein repository namens „.git“ angelegt. Es ist noch leer.**

GIT Übung – Arbeit mit dem lokalen Repository

6. Fügen Sie die Dateien Ihres Projekts dem Repository hinzu.

1. Fügen Sie die Dateien dem Staging Bereich hinzu:
   add <Dateiname1> <dateiname2> …
2. Bringen Sie den Inhalt des Staging Bereichs ins Repository
   git commit –m „neues Projekt angelegt“
3. ➔ Meldung ähnlich zu

```
[master (root-commit) fe50860] neues Projekt angelegt
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 ersteTextdatei.txt
 create mode 100644 zweiteTextDatei.txt
```

7. Ändern Sie eine der Dateien und fügen Sie eine neue Datei hinzu und löschen Sie eine Datei

GIT Übung – Arbeit mit dem lokalen Repository

8. Fragen Sie den Status ab.

git status

➔ Meldung ähnlich zu

```
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   ersteTextdatei.txt
        deleted:    zweiteTextDatei.txt

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

        dritteTextDatei.txt

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

GIT Übung – Arbeit mit dem lokalen Repository

9. **Bringen Sie Ihre Änderungen in den Stage Bereich**
   git add <geänderteDatei> <neueDatei>
   git rm <gelöschteDatei>

10. **Überprüfen Sie den Status**
    git status
11. **Bringen Sie die Änderungen ins Repository**
    git commit –message „was geandert“
12. **Betrachten Sie die Historie des Projekts**
    git log
13. **Legen Sie einen lokalen branch in Ihrem repository an**
    git branch <branchname>
14. Wec**hseln Sie in den neuen Branch**
    git checkout <branchname>

GIT Übung – Arbeit mit dem lokalen Repository

15. **Führen Sie Änderungen im neuen branch und im master durch und bringen Sie die Änderungen ins lokale repository.**
    Den jeweiligen branch auschecken git checkout <branch_name>
    Änderungen im jewieligen branch machen, dann mit git add und git commit ins repository bringen.

16. **Mergen Sie den neuen brach in den master branch.**
    Wechseln in den master (falls nicht ohnenhin schon passiert):
    git checkout master
    dann mergen: git merge <branch_name>

17. **Löschen Sie den branch**
    git branch –d <branch_name>

Git Übung - Zusammenarbeit

1. **Klonen Sie Ihr erstes Repository:**
   git clone <Pfad zum ersten Repository> <Pfad zum zu erstellenden Klon>
2. **Betrachten Sie die entstandene Folder Struktur:**
   unter dem Pfad des Klons liegt ein Abbild des Originals, incl eines git Repositories.
3. **Ändern Sie eine Datei im ursprünglichen Repository und bringen Sie die Änderung ins Repository**
   1. Datei ändern
   2. Datei in den Staging Bereich bringen
   3. Committen
4. **Ändern Sie eine andere Datei im geklonten repository**
   1. Datei ändern
   2. Datei in den Stage Bereich bringen
   3. Committen

Git Übung - Zusammenarbeit

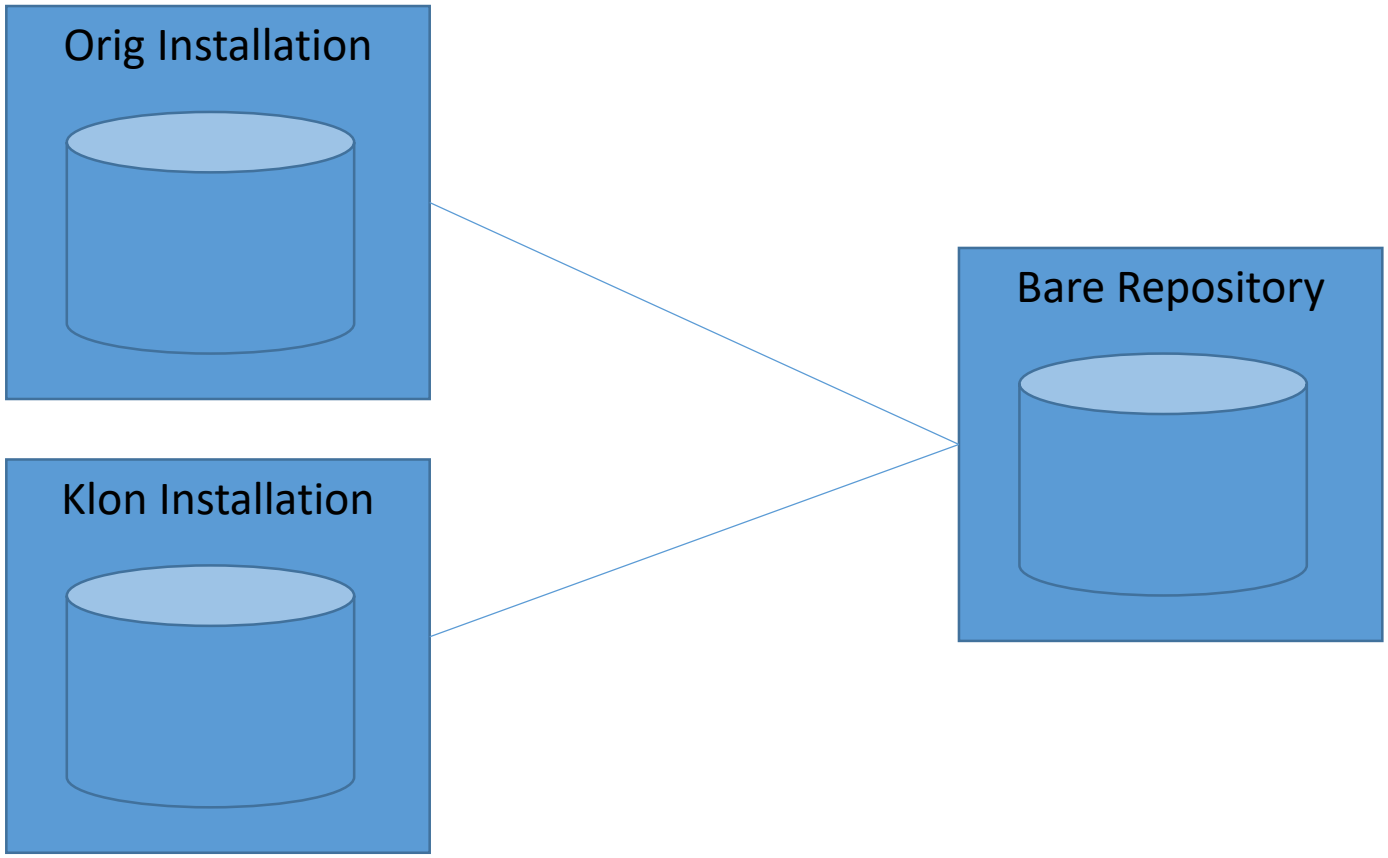
5. **➔ Aktueller Stand: Unterschiedliche Änderungen in den lokalen Repositories.**
6. **Holen Sie die Änderungen aus dem Original in den Klon:**
   im Verzeichnis des Klons:
   git pull
7. **Lassen Sie sich das log graphisch anzeigen:**
   git log –graph
   ➔ nichtlineare Historie
8. **Aktueller Stand: im Klon sind alle Änderungen, im Original befinden sich nur die Änderungen aus dem Original**
9. **Holen Sie die Änderungen aus dem Klon ins Original (der Pull Befehl ohne Argument funktioniert nur bei geklonten Repositories)**
   im Verzeichnis des Originals:
   git pull <pfad zum klon> master

Git Übung – Zusammenarbeit über bare repository

1. **Erstellen Sie ein repository ohne Workspace als klon des Original Repositories. (ein bare repository verwendet die Endung .git)**
   git clone –bare <Pfad zum original> <pfad zum neuen bare repository>
   Bsp: git clone projekte/important_project projekte/austausch_repo.git
2. **Ändern Sie eine Datei im Original, z.B. ersteTextDatei.txt und bringen Sie die Änderung ins Repository**
3. **Pushen Sie die Änderung ins Austausch repository**
   git push <Pfad zum Austausch repository> master
4. **Holen Sie die Änderungen aus dem austausch Repository in den Klon**
   git pull <Pfad zum Austausch Repository> master
5. **Überprüfen Sie die Historie und den Stand der Files.**

Git Übung – Erweiterung: Branches

Sie möchten an einem branch gemeinsam mit Ihrem Mitarbeiter über das zentrale (bare) repository arbeiten.



Git Übung – Erweiterung: Branches

Sie möchten an einem branch gemeinsam mit Ihrem Mitarbeiter über das zentrale (bare) repository arbeiten.

1. **Erstellen Sie einen branch in Ihrem repository mit dem Namen „develop“.**
   Im branch master: git branch develop
2. **Wechseln Sie in den branch develop und machen Sie Änderungen, die Sie ins repository schreiben.**
   git checkout develop
   Änderungen machen
   git add <file_name>
   git commit –m „<message>“
3. **Bringen Sie den neuen branch in das bare repository.**
   git push <Pfad zum bare repository> develop

Git Übung – Erweiterung: Branches

4. **Gehen Sie in die Klon-Installation.**
5. **Fügen Sie das bare repository als remote repository hinzu.**
   git remote add <lokaler remote name> <Pfad zum remote repo>
6. **Holen Sie die Änderungen aus dem bare repository zu sich ins Klon-repository.**
   git fetch <lokaler remote name>
7. **Prüfen Sie die remote branches in Ihrem Klon repo**
   git remote –v
8. **Erstellen Sie einen lokalen branch basierend auf dem neuen remote develop-branch und wechseln Sie in den branch**
   git fetch <lokaler remote name>
   git checkout –b develop <lokaler remote name> /<branchname>
9. **Arbeiten Sie auf dem neuen lokalen branch und bringen Sie die Änderungen ins bare repository.**

Material

Zu verwendendes Material:

- VL Skript
- ProGit
- (R. Preißel, B. Stachmann: Git)