



Faculty of Computer Science, Institute of Systems Architecture, Chair of Systems Engineering

RoboLab Spring Course 2020

Git - Crashkurs

von M.Sc. Samuel Knobloch

Gliederung

- Git – Ein (sehr) kleiner Überblick
- Wichtige Befehle
- Wenn man weiß, was man tut...
- Alternative: IDE

Git – Ein (sehr) kleiner Überblick

Git – Ein (sehr) kleiner Überblick

- Verteiltes Version Control System (DVCS)
 - Lokale und entfernte Repositories möglich
- Geeignet für individuelles Arbeiten und Synchronisieren in Teams
 - Paralleles Entwickeln mehrerer Versionen
- Bietet Historie und Zurücksetzen auf ältere Stände
- Automatisches Zusammenführen unterschiedlicher Stände

Wichtige Befehle

Wichtige Befehle

- Lokal neues Repository anlegen

```
$ cd your/target/directory
```

```
$ git init
```

Wichtige Befehle

- Repository von Server klonen

```
$ git clone git@remote.url:repo/path <dir>
```

```
$ git clone https://remote.url/repo/path <dir>
```

- Erweiterbar um Flags
 - Wichtiges Flag: `--recursive`

Wichtige Befehle

- Neue Dateien hinzufügen

```
$ git add <file>  
$ git add *.<type>  
$ git add .
```


Wichtige Befehle

- Status der lokalen Dateien prüfen

```
$ git status
```

Wichtige Befehle

- Nach dem Hinzufügen Commit erstellen

```
$ git commit <file> <path> ... -m "<message>"
```

```
$ git commit -a -m "<message>"
```

Wichtige Befehle

- Remote hinzufügen

```
$ git remote add origin git@remote.url:repo/path
```

```
$ git remote add origin https://remote.url/repo/path
```

Wichtige Befehle

- Informationen vom Remote abfragen

```
$ git fetch
```

```
$ git fetch -p
```

Wichtige Befehle

- Remote mit Lokalen Stand synchronisieren

```
$ git push -u origin <branch>
```

- Standard-Branch heißt master

Wichtige Befehle

- Lokalen Stand mit Remote synchronisieren

```
$ git pull
```

- Kann u.U. einen Merge erzeugen

Wichtige Befehle

- Branching

```
$ git branch <new_branch>  
$ git checkout <new_branch>  
  
$ git checkout -b <new_branch>
```

Wenn man weiß, was man tut...

Wenn man weiß, was man tut...

- Zwei Branches zusammenführen

```
$ git checkout <first_branch>  
$ git merge <other_branch>  
  
$ git commit -a -m "Merged something"
```

Wenn man weiß, was man tut...

- Mergen ohne Commit zu erzeugen

```
$ git checkout <first_branch>  
$ git rebase -i origin/<other_branch>  
  
$ git push --force
```

Wenn man weiß, was man tut...

- Der bessere Pull-Befehl

```
$ git pull --rebase
```

Alternative: IDE

Alternative: IDE

- Moderne IDEs bieten gute bis sehr gute Integration
 - Git, Mercurial, Subversion, CVS
- Vieles wird weg gekapselt, Abläufe sind nicht sichtbar
 - Für Verständnis nicht immer gut
- Für Anfänger schwieriger bei Fehlern (die garantiert auftreten werden)