

# Arbeiten mit Git

2014-01-28

Georg Wirtenberger

# Inhalt

1. Einleitung
2. Grundlagen
  - ▶ Versionsverwaltung
  - ▶ Snapshots
  - ▶ Drei Zustände
  - ▶ Dateistatus
  - ▶ Branches
3. Arbeiten mit Git
  - ▶ Arbeiten am Clone
  - ▶ Arbeiten am Master
  - ▶ Arbeiten an Branches (Zweigen)
  - ▶ Repository und Clone zusammenführen (mergen)
4. Git-Workflows
  - ▶ Integration-Manager-Workflow
5. Quellen

# Einleitung

- ▶ kurzer Einblick in Git
- ▶ nur die wichtigsten Grundlagen, die zum Arbeiten und Verstehen nötig sind
- ▶ mehr über Git siehe Quellen

# Grundlagen

## Versionsverwaltung

- ▶ arbeiten an Dateien ohne Verlust von älteren Versionen
- ▶ arbeiten an einem Projekt mit mehreren Personen
- ▶ SVN und Git
- ▶ Git arbeitet intern anders als SVN

# Grundlagen

## Snapshots

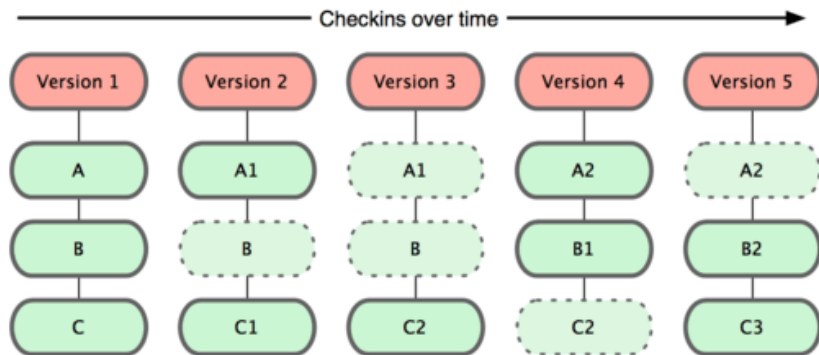


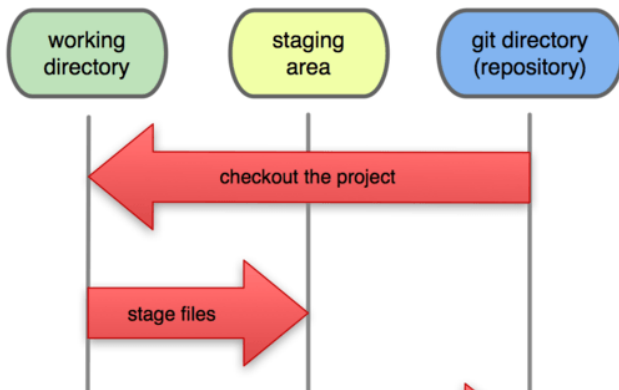
Figure 1: Snapshots über die Zeit

# Grundlagen

## Drei Zustände

- ▶ eingecheckt (committed), vorbereitet (staged) und verändert (modified)

### Local Operations

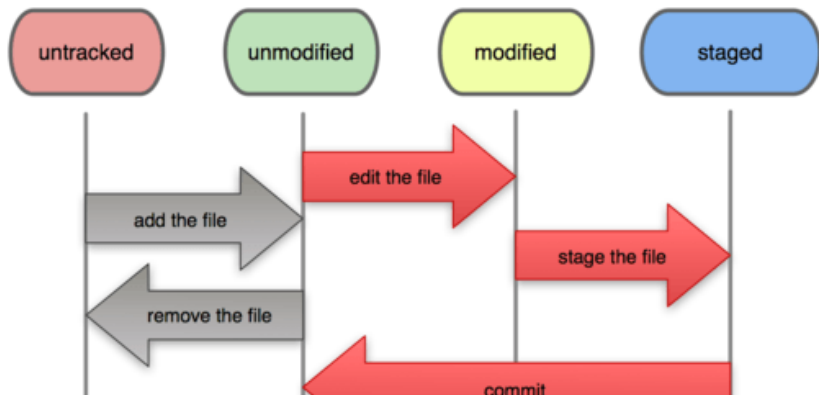


# Grundlagen

## Dateistatus

- entweder verfolgt(tracked) oder oder nicht verfolgt (untracked).

### File Status Lifecycle



# Grundlagen

## Branches

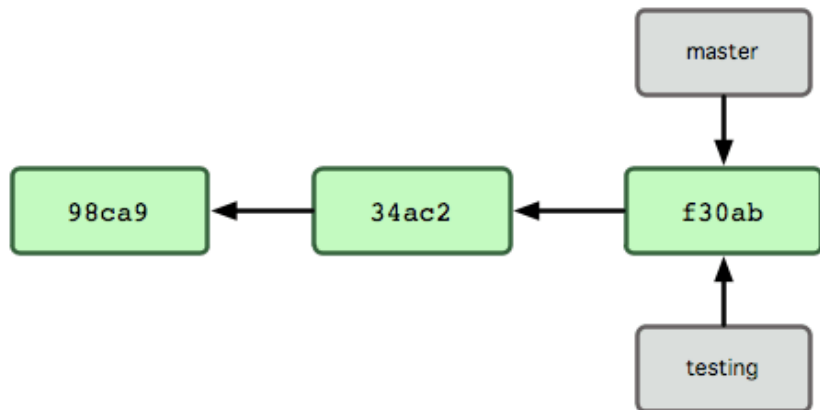


Figure 4: Beispiel einer Verzweigung



# Arbeiten mit Git

## Erstellen und Clonen

- ▶ Repository erstellen

```
$ git init
```

- ▶ Repository clonen

```
$ git clone git://github.com/topaz/topaz.git
```

# Arbeiten mit Git

## Arbeiten am Clone

- ▶ Arbeiten am Master
- ▶ Arbeiten am Branch

# Arbeiten mit Git

## Arbeiten am Master

- ▶ Sie bearbeiten die Daten
- ▶ Sie fügen die daten zum Index hinzu (track)

```
$ git add *.js
```

- ▶ Sie entfernen die Daten aus der Repository

```
$ rm test.js
```

- ▶ Sie cheken die Daten ein (commiten)

```
$ git commit -m 'alle Javascriptdateien wurden eingeche
```

- ▶ Sie wechseln zur vorherigen Version

```
$ git checkout [Hashwert des Commits]
```

# Arbeiten mit Git

## Arbeiten an Branches (Zweigen)

- ▶ Sie erstellen ein Branch

```
$ git branch myBranch
```

```
$ git checkout myBranch
```

- ▶ Sie bearbeiten die Daten
- ▶ Sie fügen die Daten zum Index hinzu (track)

```
$ git add *.js
```

- ▶ Sie entfernen die Daten aus der Repository

```
$ rm test.js
```

- ▶ Sie cheken die Daten ein (commiten)

```
$ git commit -m 'alle Javascriptdateien wurden eingeche
```

# Arbeiten mit Git

## Arbeiten an Branches (Zweigen)

- ▶ Sie wechseln zur vorherigen Version

```
$ git checkout [Hashwert des Commits]
```

- ▶ Sie führen Branch mit Master zusammen (mergen)

```
$ git checkout master
```

```
$ git merge myBranch
```

- ▶ Sie löschen den Branch

```
$ git branch -d myBranch
```

# Arbeiten mit Git

## Repository und Clone zusammenführen (mergen)

- ▶ `$ git push origin master`
- ▶ `$ git pull`

# Git-Workflows

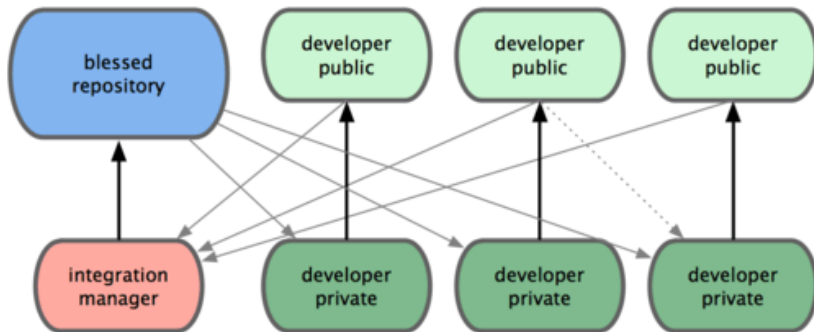


Figure 5: Integration-Manager Workflow

# Quellen

- ▶ Pro Git: <http://git-scm.com/book/de/>