

Übung zu Betriebssystembau

Git-Crashkurs

November 2022

Peter Ulbrich & Alexander Lochmann
(Mit Material vom Lehrstuhl 4 der FAU)

Arbeitsgruppe Systemsoftware
Technische Universität Dortmund

Eine kleine Einführung in Versionsverwaltungssysteme

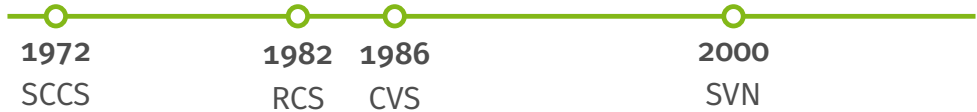


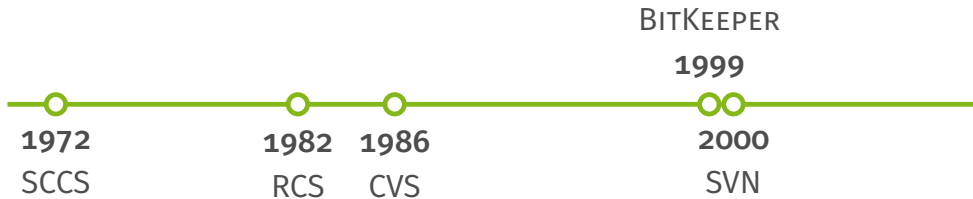
1972

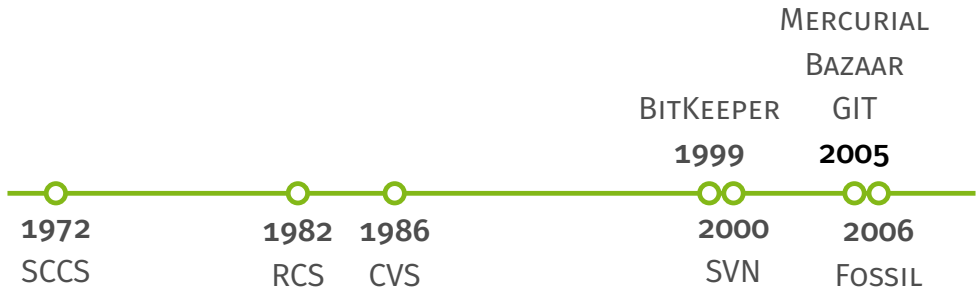
SCCS

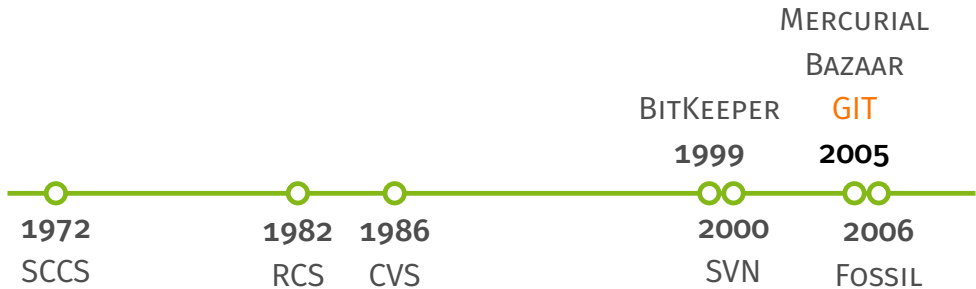












Schlüsselkonzepte von GIT

- nicht-lineare Entwicklung (*branch*)
- Integrität durch SHA-1 Hash
- vollständiges Speichern der Daten (*snapshot*)
- dezentral (*clone*)

Lokales GIT Repository initialisieren

```
heinloth:~$ mkdir beispiel  
heinloth:~$ cd beispiel
```



Workspace

Lokales GIT Repository initialisieren

```
heinloth:~$ mkdir beispiel  
heinloth:~$ cd beispiel  
~/beispiel$ git init  
Leeres Git-Repository in beispiel/.git/ initialisiert
```



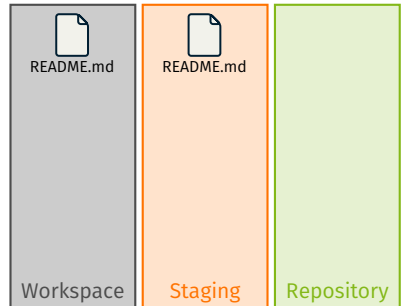
Dateien mit GIT verwalten

```
~/beispiel$ touch README.md
```



Dateien mit GIT verwalten

```
~/beispiel$ touch README.md  
~/beispiel$ git add README.md
```

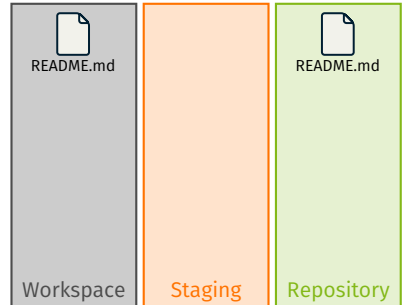


Dateien mit GIT verwalten

bd2de5c

1

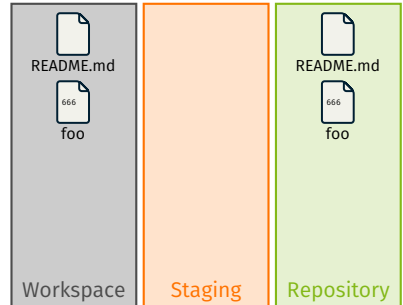
```
~/beispiel$ touch README.md
~/beispiel$ git add README.md
~/beispiel$ git commit -m "Liesmich hinzugefügt"
[master (Root-Commit) bd2de5c] Liesmich hinzugefügt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
```



Dateien mit GIT verwalten



```
~/beispiel$ echo "666" > foo
~/beispiel$ git add foo
~/beispiel$ git commit -m "Datei foo erstellt"
[master df7aa5a] Datei foo erstellt
1 file changed, 1 insertion(+)
create mode 100644 foo
```



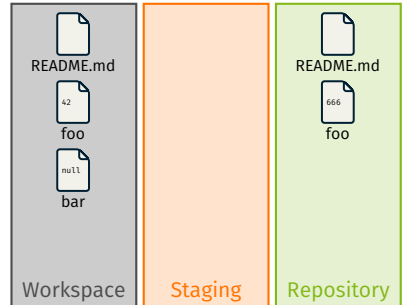
Dateien mit GIT verwalten



```
~/beispiel$ echo "42" > foo
~/beispiel$ echo "null" > bar
~/beispiel$ git status
Auf Branch master
Änderungen, die nicht zum Commit vorgemerkt sind:
  geändert: foo

Unversionierte Dateien:
  bar

keine Änderungen zum Commit vorgemerkt
```

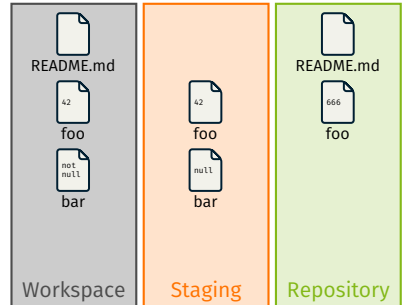


Dateien mit GIT verwalten



```
~/beispiel$ git add foo bar
~/beispiel$ echo "not null" > bar
~/beispiel$ git status
Auf Branch master
Zum Commit vorgemerkte Änderungen:
  neue Datei: bar
  geändert: foo

Änderungen, die nicht zum Commit vorgemerkt sind:
  geändert: bar
```

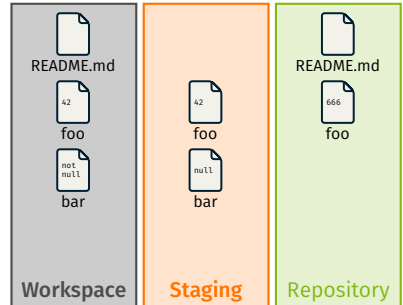


Dateien mit GIT verwalten

master



```
~/beispiel$ git diff
diff --git a/bar b/bar
index 19765bd..b263a85 100644
--- a/bar
+++ b/bar
@@ -1,1 @@
-null
+not null
```

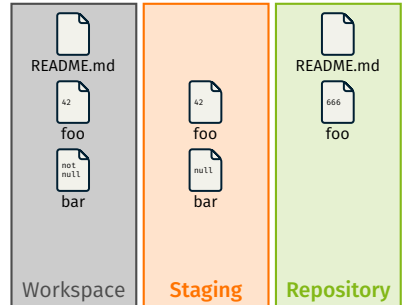


Dateien mit GIT verwalten

master



```
~/beispiel$ git diff --staged
diff --git a/bar b/bar
new file mode 100644
index 0000000..19765bd
--- a/bar
+++ b/bar
@@ -0,0 +1 @@
+null
diff --git a/foo b/foo
index 7cc86ad..d81cc07 100644
[...]
```

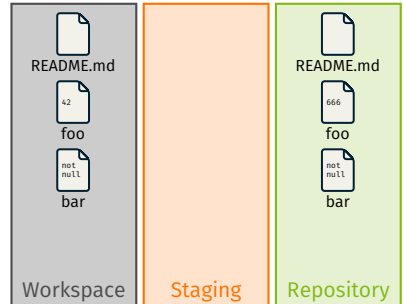


Dateien mit GIT verwalten

master



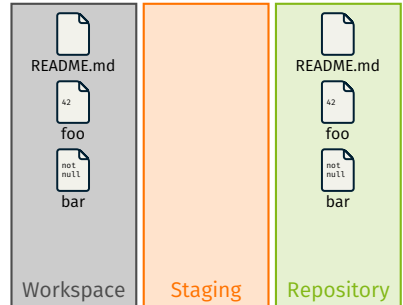
```
~/beispiel$ git add bar
~/beispiel$ git commit -m \
    "Foo korrigiert und Bar erstellt"
[master 90f7cfe] Foo korrigiert und Bar erstellt
2 files changed, 2 insertions(+), 1 deletion(-)
create mode 100644 bar
```



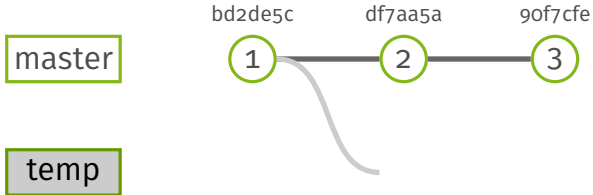
Dateien mit GIT verwalten



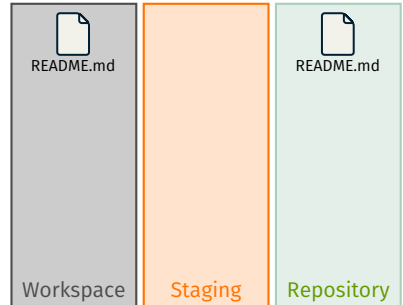
```
~/beispiel$ git shortlog
Bernhard Heinloth (3):
  Liesmich hinzugefügt
  Datei foo erstellt
  Foo korrigiert und Bar erstellt
```



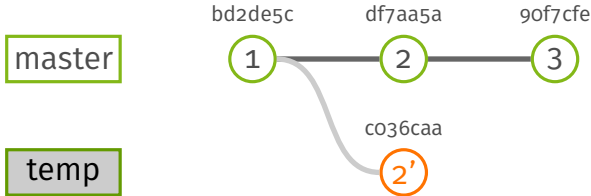
GIT Zweige



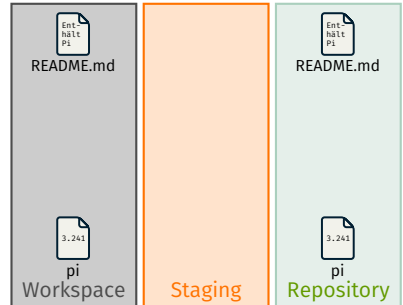
```
~/beispiel$ git branch temp bd2de5c
~/beispiel$ git checkout temp
Zu Zweig »temp« gewechselt
~/beispiel$ git shortlog
Bernhard Heinloth (1):
  Liesmich hinzugefügt
```



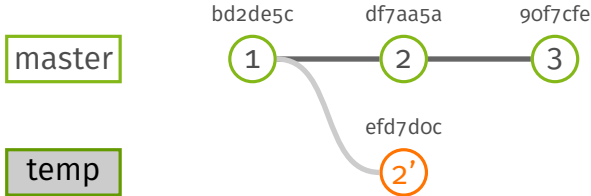
GIT Zweige



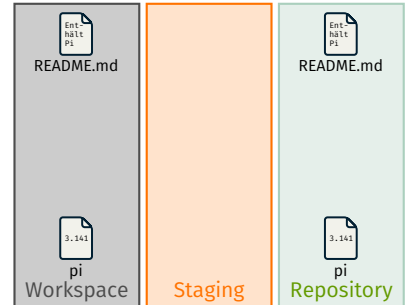
```
~/beispiel$ echo "3.241" > pi
~/beispiel$ echo "Beinhaltet Pi" >> README.md
~/beispiel$ git add .
~/beispiel$ git commit -m "Kreiszahl hinzugefügt"
[temp c036caa] Kreiszahl hinzugefügt
2 files changed, 2 insertions(+)
create mode 100644 pi
```



GIT Zweige

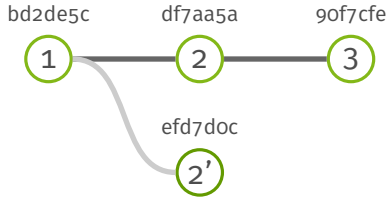


```
~/beispiel$ echo "3.141" > pi
~/beispiel$ git commit -a --amend -m \
    "Kreiszahl ergänzt"
[temp efd7d0c] Kreiszahl ergänzt
Date: Mon Oct 5 12:50:08 2020 +0200
2 files changed, 2 insertions(+)
create mode 100644 pi
```



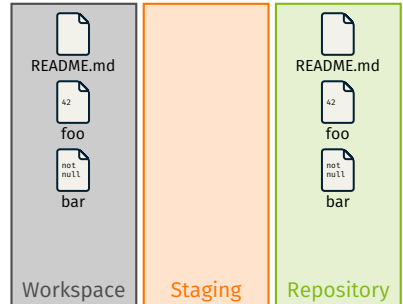
GIT Zweige

master



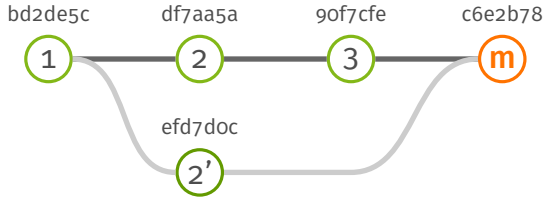
temp

```
~/beispiel$ git branch
  master
* temp
~/beispiel$ git checkout master
~/beispiel$ git branch
* master
  temp
```



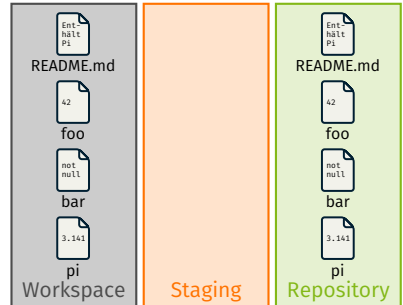
GIT Zweige zusammenführen

master



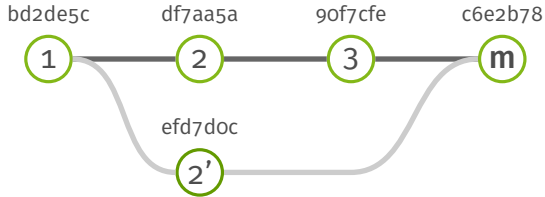
temp

```
~/beispiel$ git merge temp
Merge made by the 'recursive' strategy.
 README.md | 1 +
  pi       | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 pi
~/beispiel$ git log
commit c6e2b781a60785fa2fac0d7467b5b0e7c9a7fb8c
Merge: 90f7cfe efd7d0c
Author: Bernhard Heinloth <heinloth@cs.fau.de>
[...]
```



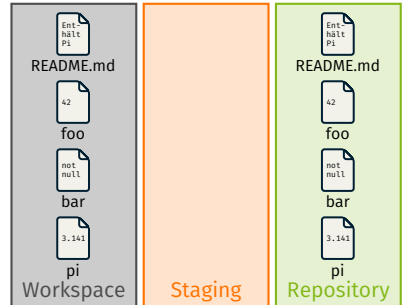
GIT Zweige zusammenführen

master

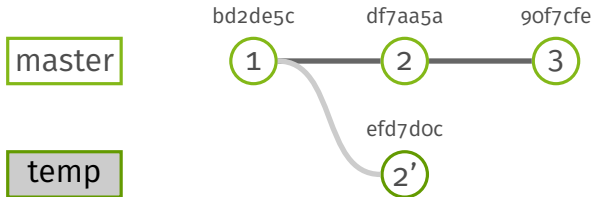


temp

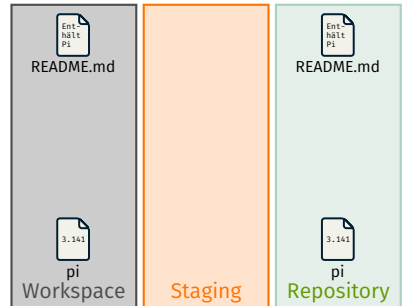
```
~/beispiel$ git shortlog
Bernhard Heinloth (5):
  Liesmich hinzugefügt
  Datei foo erstellt
  Foo korrigiert und Bar erstellt
  Kreiszahl ergänzt
  Merge branch 'temp'
```



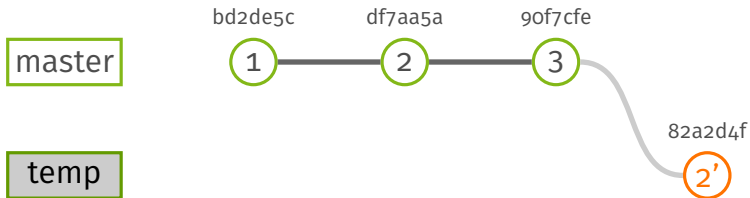
Alternativ: Die GIT Geschichte neu schreiben



```
~/beispiel$ git branch
  master
* temp
```

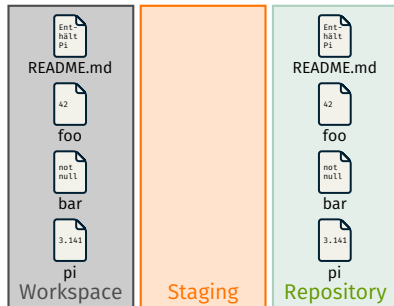


Alternativ: Die GIT Geschichte neu schreiben

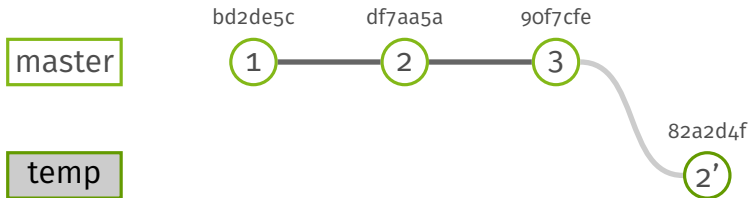


```
~/beispiel$ git rebase master
Zunächst wird der Branch zurückgespult,
um Ihre Änderungen darauf neu anzuwenden...
Wende an: Kreiszahl ergänzt
~/beispiel$ git log
commit 82a2d4f28d9986560aa75ea429ac5f510eebfd99
Merge: 90f7cfe efd7d0c
Author: Bernhard Heinloth <heinloth@cs.fau.de>
Date: Mon Oct 5 12:50:08 2020 +0200
```

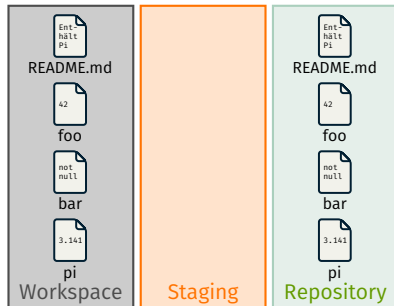
Kreiszahl ergänzt



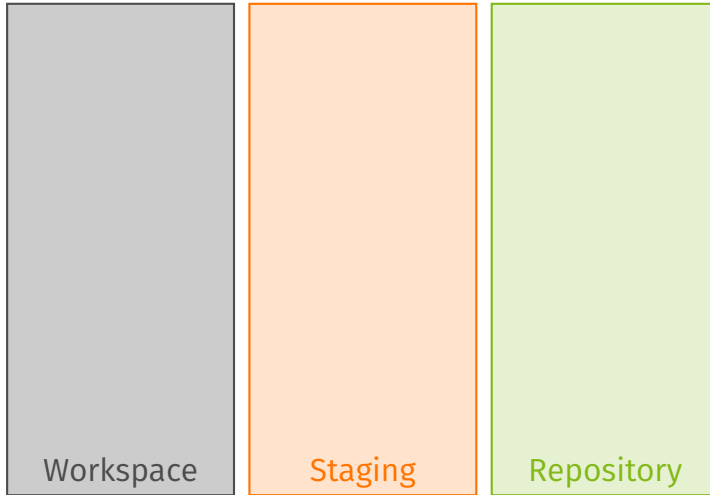
Alternativ: Die GIT Geschichte neu schreiben



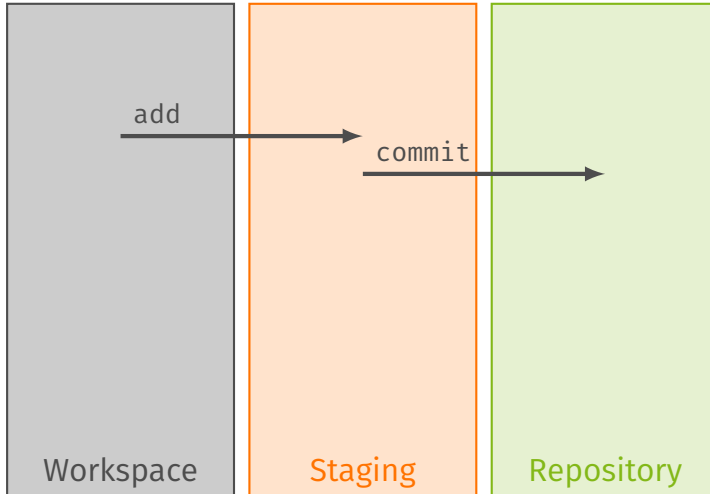
```
~/beispiel$ git shortlog
Bernhard Heinloth (4):
  Liesmich hinzugefügt
  Datei foo erstellt
  Foo korrigiert und Bar erstellt
  Kreiszahl ergänzt
```



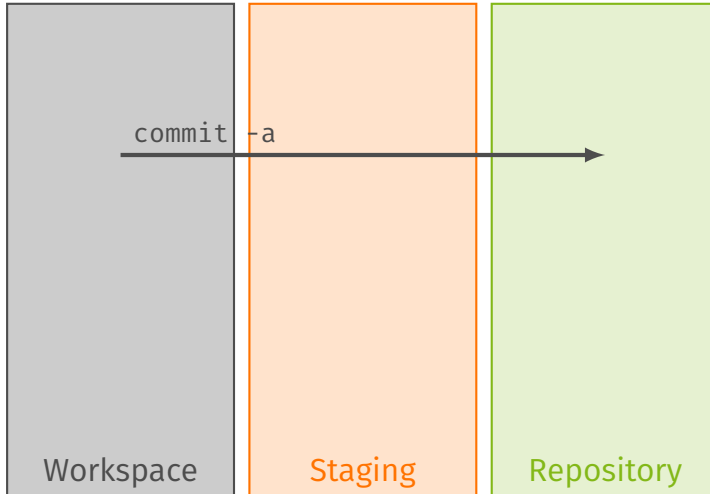
Überblick: Dateien in GIT ein- und auschecken



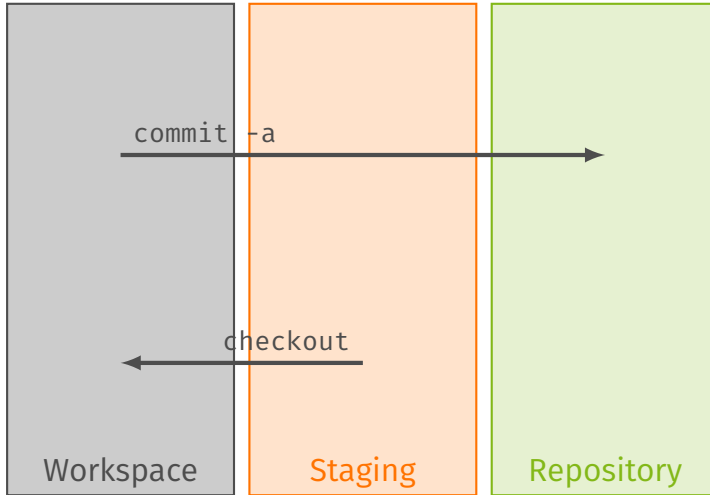
Überblick: Dateien in GIT ein- und auschecken



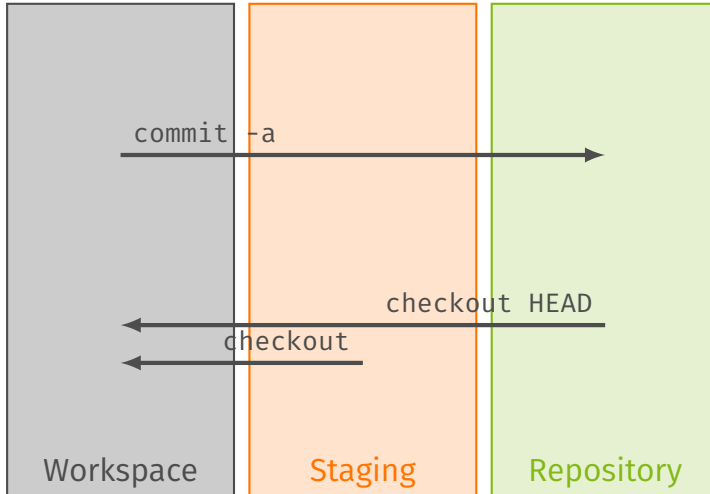
Überblick: Dateien in GIT ein- und auschecken



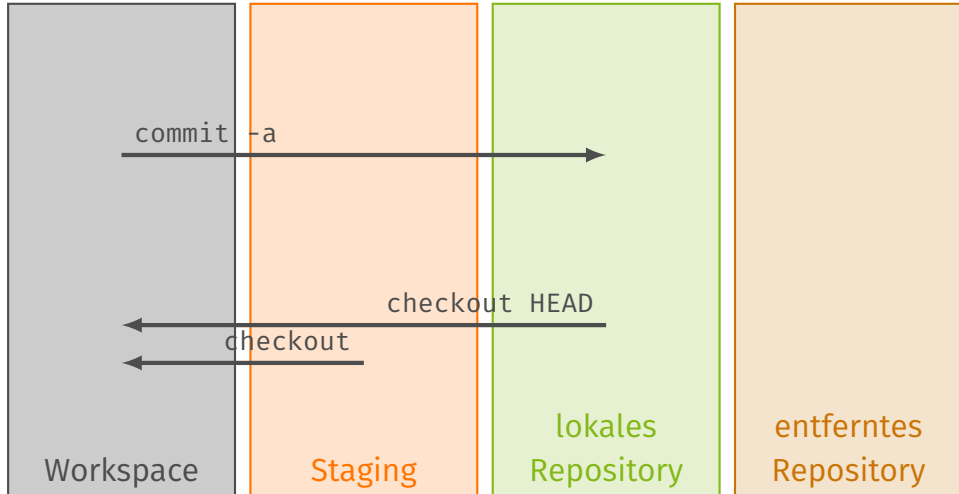
Überblick: Dateien in GIT ein- und auschecken



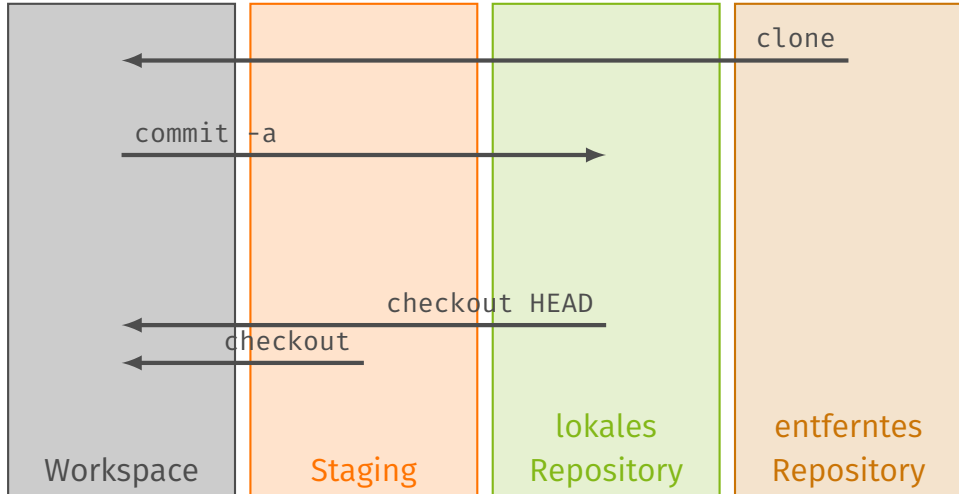
Überblick: Dateien in GIT ein- und auschecken



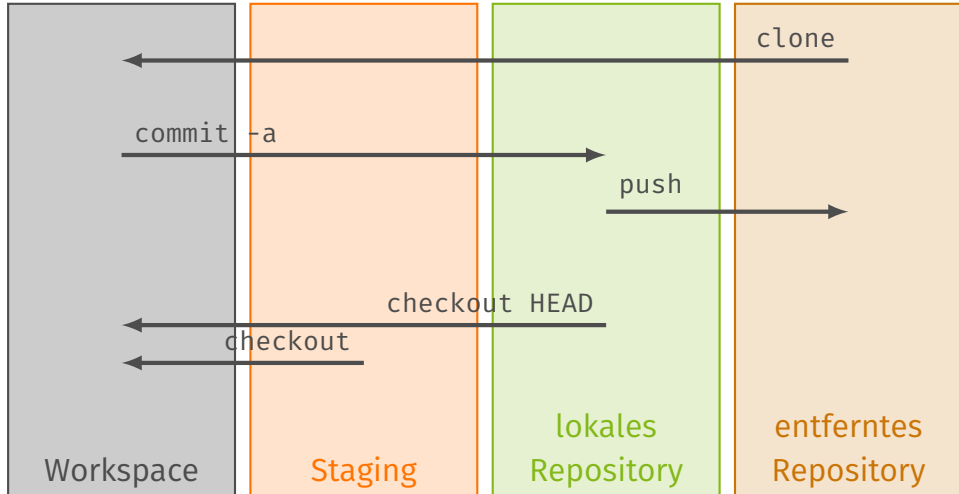
Überblick: Dateien in GIT ein- und auschecken



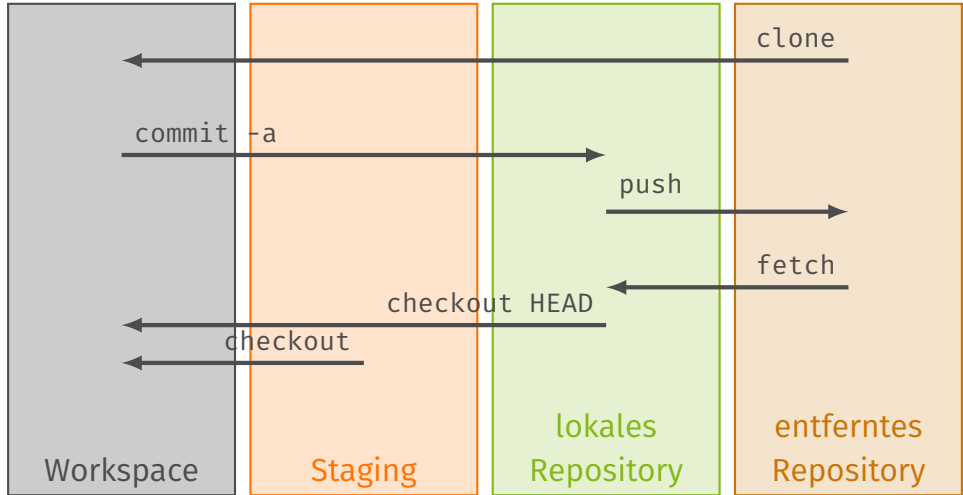
Überblick: Dateien in GIT ein- und auschecken



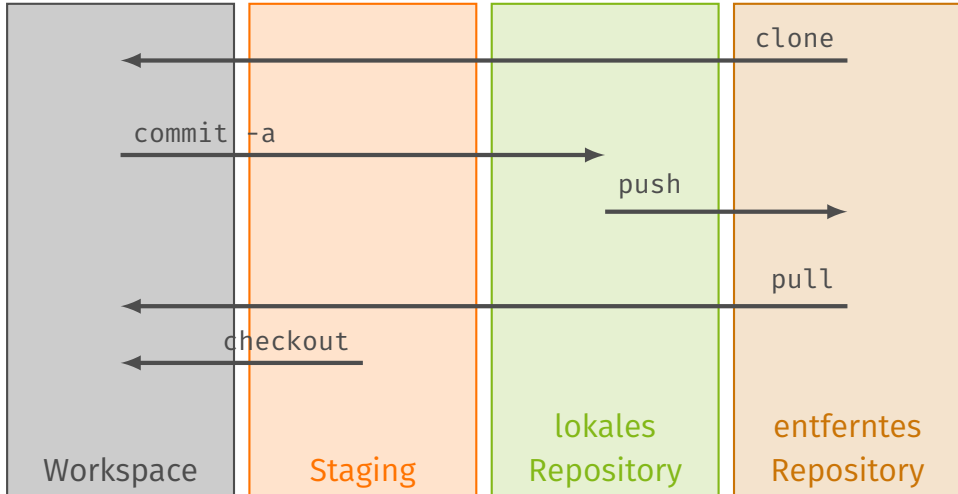
Überblick: Dateien in GIT ein- und auschecken



Überblick: Dateien in GIT ein- und auschecken



Überblick: Dateien in GIT ein- und auschecken



- Viele (kommerzielle) Anbieter, meist mit Weboberfläche zur Verwaltung:



- Eigene GITEA-Instanz des Departments Informatik auf `git.cs.tu-dortmund.de`
 - erlaubt kostenlos private Repos
 - unterstützt *Continuous Integration* (CI)
 - keine Registrierung notwendig, Anmeldung über *IRB-Account*, bitte jede(r) einmal anmelden
- GITEA Übungsrepo wird automatisch nach Anmeldung erstellt

Workflow

STUBS Vorlage



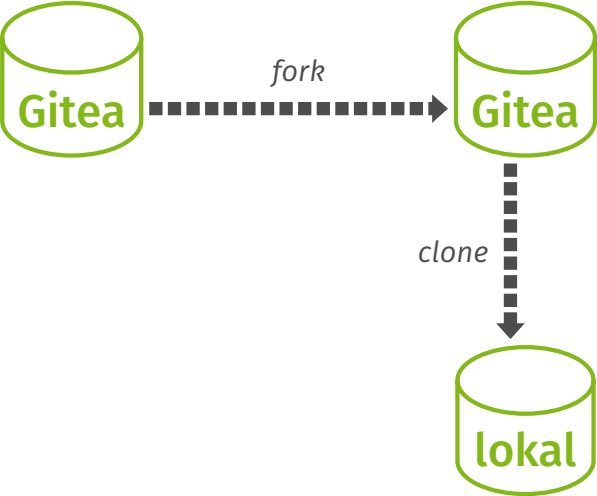
STuBS Vorlage

Repository der Gruppe



STuBS Vorlage

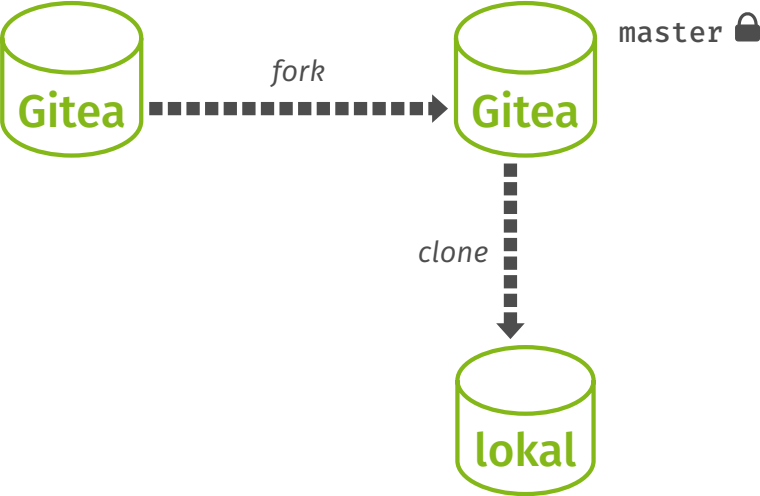
Repository der Gruppe



Arbeitskopie

STuBS Vorlage

Repository der Gruppe



Arbeitskopie

STUBS Vorlage

Repository der Gruppe



fork



master 🔒

clone



master
↓ *branch*
aufgabe-X

Arbeitskopie

STuBS Vorlage

Repository der Gruppe



fork



master 🔒

clone



aufgabe-X

Arbeitskopie

STuBS Vorlage

Repository der Gruppe



fork



master 🔒

aufgabe-X

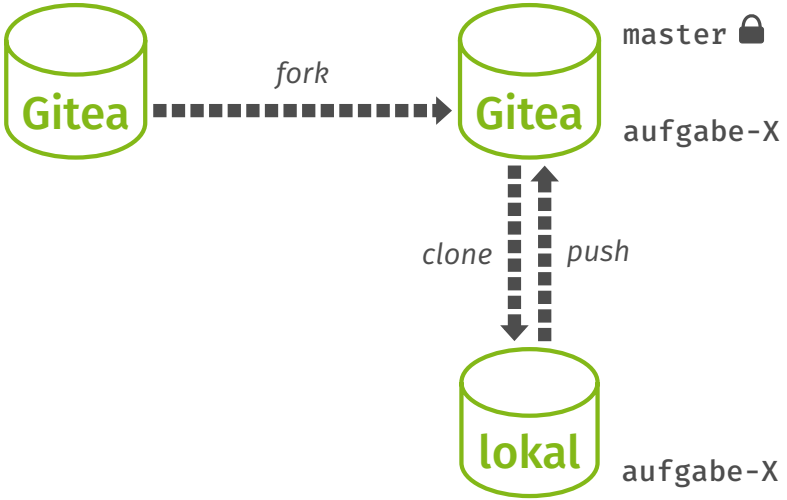
clone

push



aufgabe-X

Arbeitskopie



STuBS Vorlage

Repository der Gruppe



fork



master 

 Merge Request

aufgabe-X

clone

push

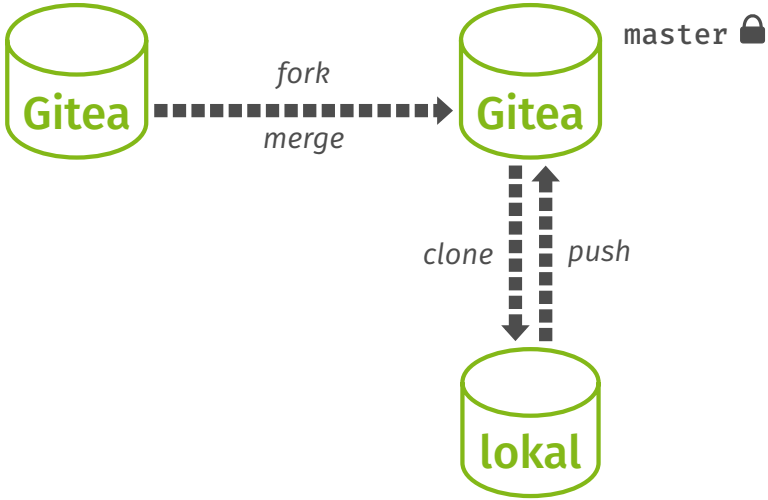


aufgabe-X

Arbeitskopie

STuBS Vorlage

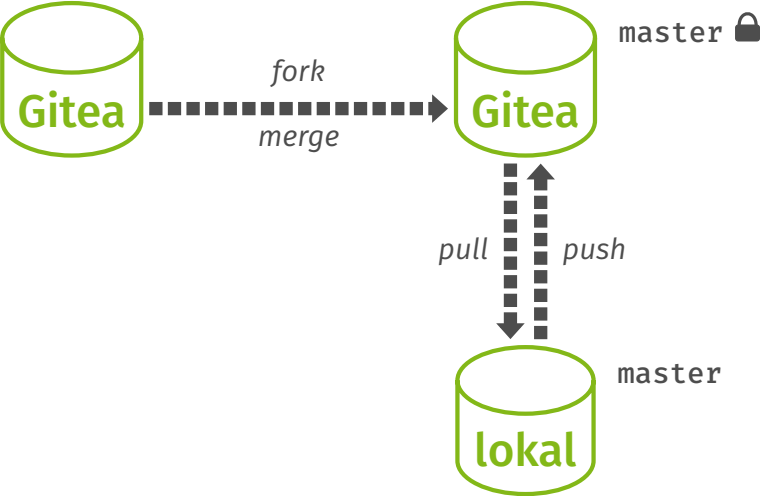
Repository der Gruppe



Arbeitskopie

STuBS Vorlage

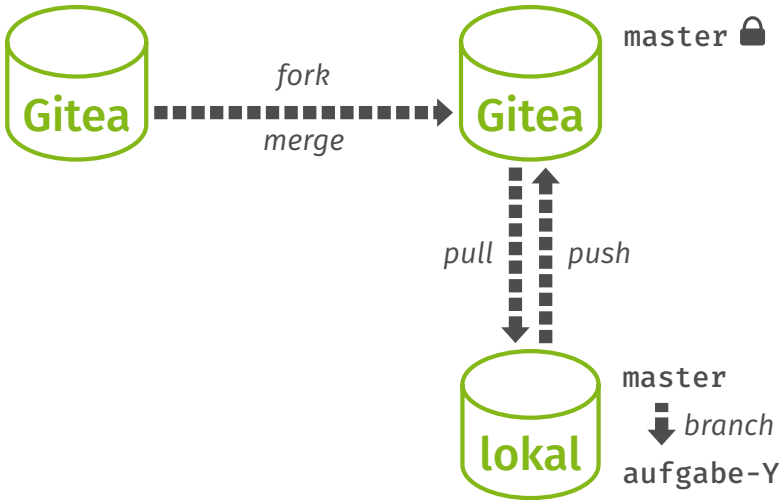
Repository der Gruppe



Arbeitskopie

STuBS Vorlage

Repository der Gruppe



Arbeitskopie

- Name und Mailadresse setzen

```
$ git config --global user.email max.mustermann@tu-dortmund.de  
$ git config --global user.name "Max Mustermann"
```

- GIT Werkzeuge anpassen

```
$ git config --global core.editor nano  
$ git config --global merge.tool meld  
$ git config --list
```

- *Optional:* Aliase definieren

```
$ git config --global alias.word-diff=diff --word-diff=color -b  
$ git word-diff
```

- *Optional:* SSH-Key erstellen und öffentlichen Schlüssel in GITEA eintragen

Arbeiten mit entferntem GITEA Repository

- Entferntes GITEA Übungsrepo der Gruppe XX (lokal) klonen

```
$ git clone git@gssh://git@git.cs.tu-dortmund.de:2222/BSB-WS22/gruppe-XX.git
```

- Neue Änderungen (*commits*) in GITEA kopieren

```
$ git push
```

Ggf. Zweig aufgabe-X in GITEA anlegen

```
$ git push --set-upstream origin aufgabe-X
```

- Änderungen aus GITEA laden

```
$ git pull
```

Bei Problemen (*merge conflict*) mit Werkzeug (hier: MELD) lösen

```
$ git mergetool --tool=meld
```




- git init** neues Repository im aktuellen Verzeichnis erstellen
- git add *Datei*** Datei als Kandidat für den nächsten *commit* markieren
- git commit** Änderungen versionieren
- git diff** unversionierte Änderungen anzeigen
- git show** neuste (versionierte) Änderungen anzeigen
- git status** Änderungen zum Vorgänger anzeigen
- git branch** verfügbare Zweige anzeigen
- git log** Historie anzeigen
- man git-Option** Hilfe anzeigen, z.B. man `git-add`

Cheatsheet (entfernte Quellen)

git clone *URL* initiales Kopieren von einer Quelle

git fetch *Name* Änderungen aus entfernter Quelle holen

git pull *Name* kurz für holen und zusammenfügen

git checkout *Zweig* Aktuellen Zweig wechseln

git push *Name* in entfernte Quelle übertragen