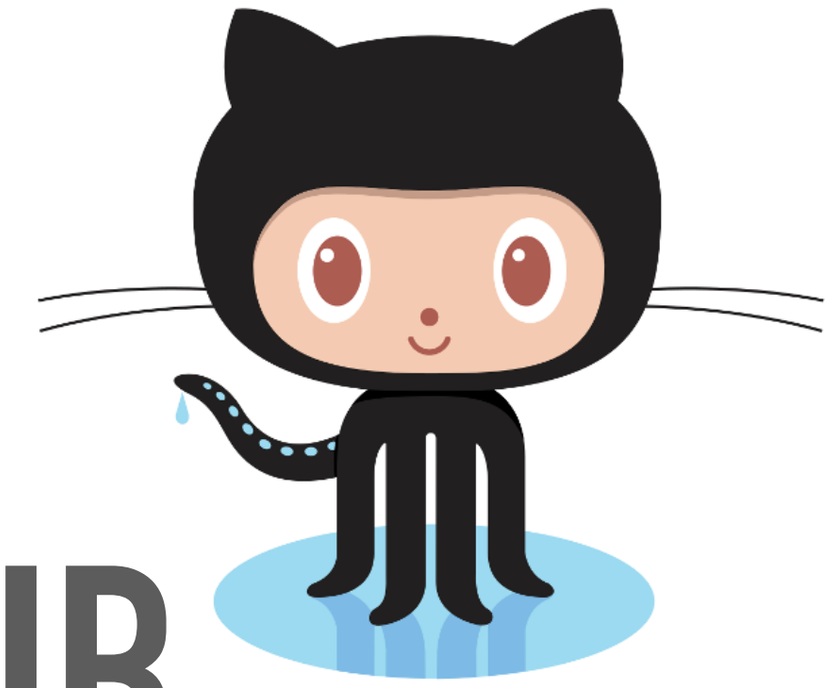


GIT / GITHUB

SEW
DI Thomas Helml

SJ 2019/20

WELS

INHALT

- Motivation
- Arten von VCS
- Git Grundlagen
- GitHub
- Git Shell Grundlagen

INHALT

- Motivation
- Arten von VCS
- Git Grundlagen
- GitHub
- Git Shell Grundlagen

WOZU VCS?

- VCS (Version Control System)
  - dt. Versionsverwaltungssystem
  - Erlaubt Teamarbeit – mehrere Leute arbeiten an der gleichen Codebasis
  - Art „Zeitmaschine“ – Backup der Entwicklungsschritte
  - Ermöglichen CI (Continuous Integration)

INHALT

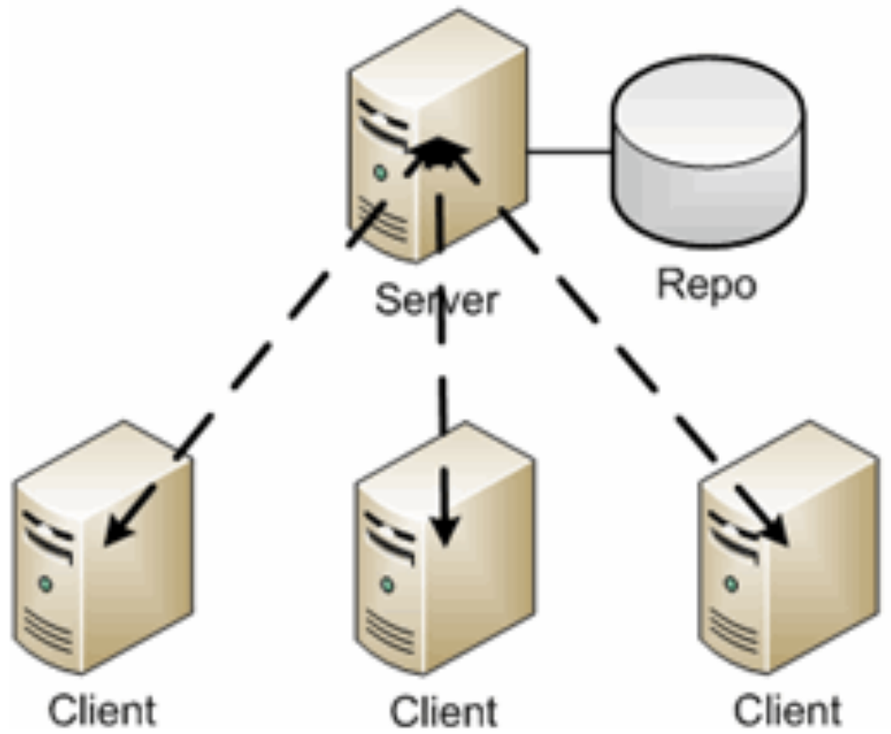
ARTEN VON VCS

- Local
  - RCS (Mac früher)
- Centralized
  - Subversion /SVN)
  - CVS
  - Perforce
- Distributed
  - Git
  - Mercurial

ARTEN VON VCS

- Funktionsweise
- 1 Server:
  - Speichert alle Versionen der Files
- viele Clients
  - Client checkt lokale Kopie aus
  - Bearbeitet sie
  - Pusht sie an Server zurück

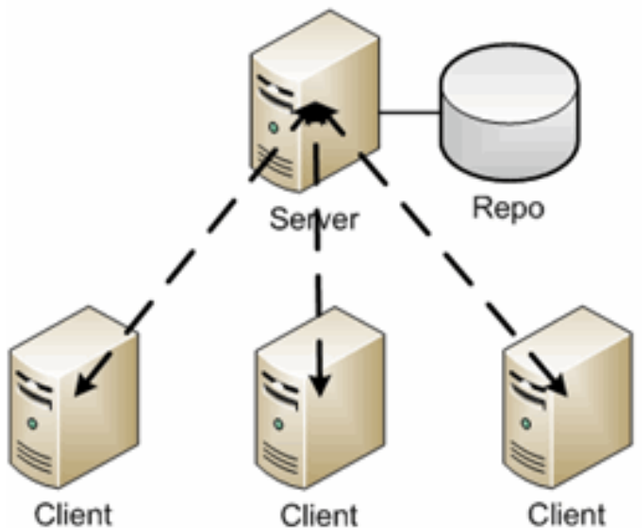
Centralized



ARTEN VON VCS

- **Vorteile**
  - schaut simpel aus, ist simpel
  - gute Verbreitung, gute IDE Integration
  - funktioniert
- **Nachteile**
  - Man kann nicht offline commiten
  - Man kann nicht mehrere Repositories in ein Projekt einbinden
  - Schwierigkeiten bei großen Teams (Open Source)

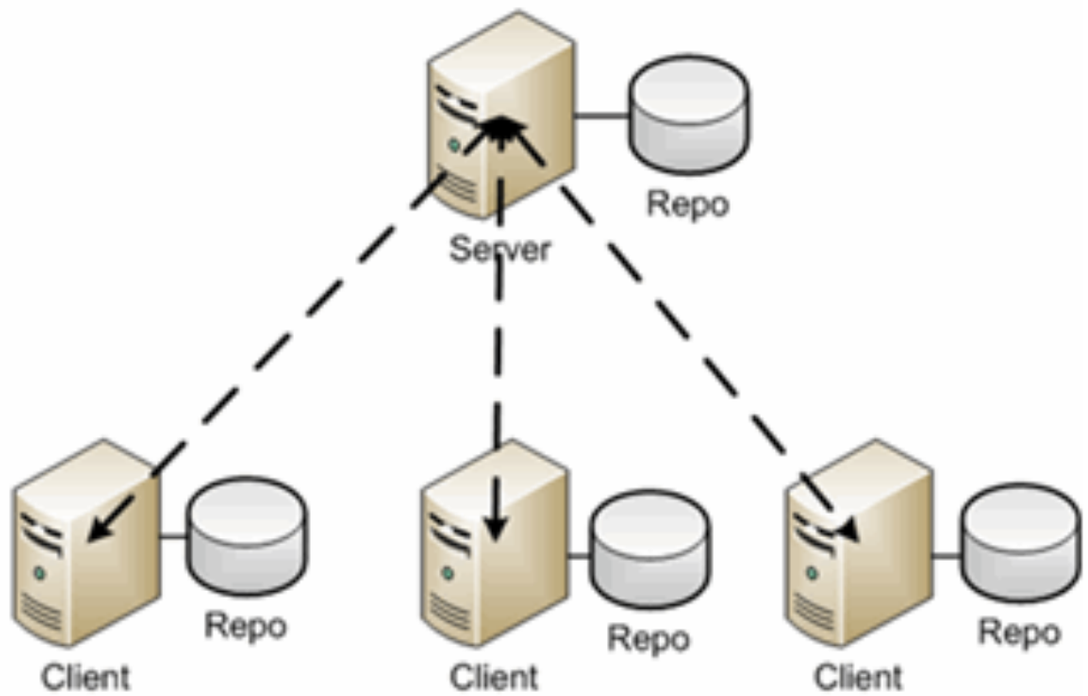
Centralized



ARTEN VON VCS

- Ein (mehrere) Server
- Client hat volle Kopie der Repository lokal, die mit Server gemerged werden können

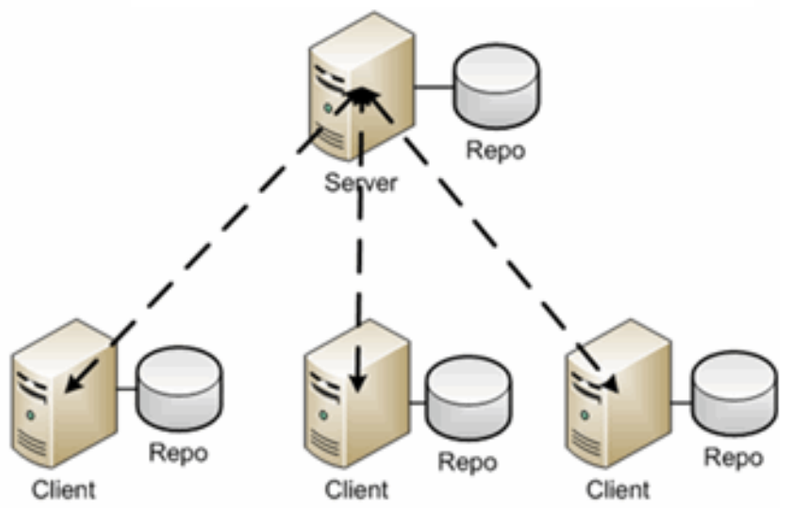
Distributed



ARTEN VON VCS

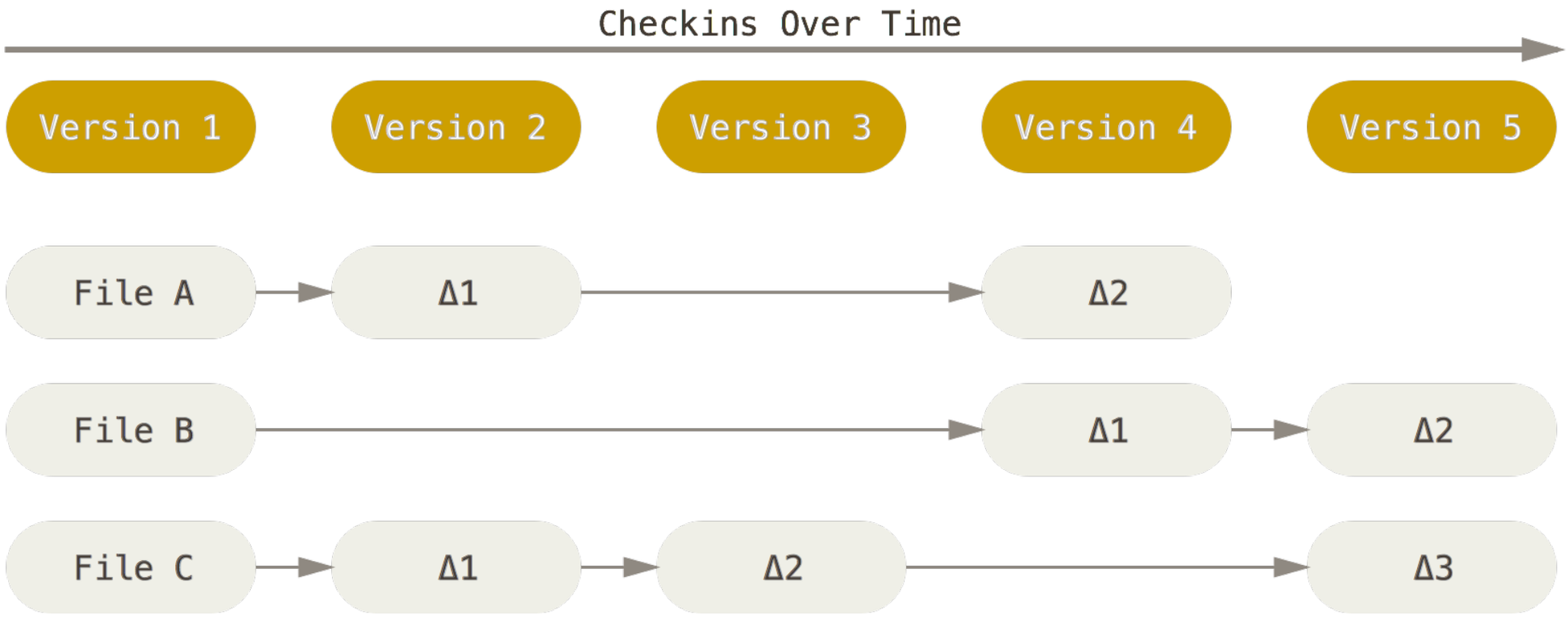
- Vorteile:
  - Kein Netzwerk notwendig/Offline möglich
  - Private Arbeit möglich ohne Muss des publishen
  - Kann beliebig viele Repos einbinden
  - Kein Single Point of Failure
  - Zentrale Kontrolle von „Server Releases“ möglich
  - Client hat volle Kopie der Repository lokal
- Nachteile:
  - Schwieriger zu lernen
  - Komplexerer Workflow

Distributed



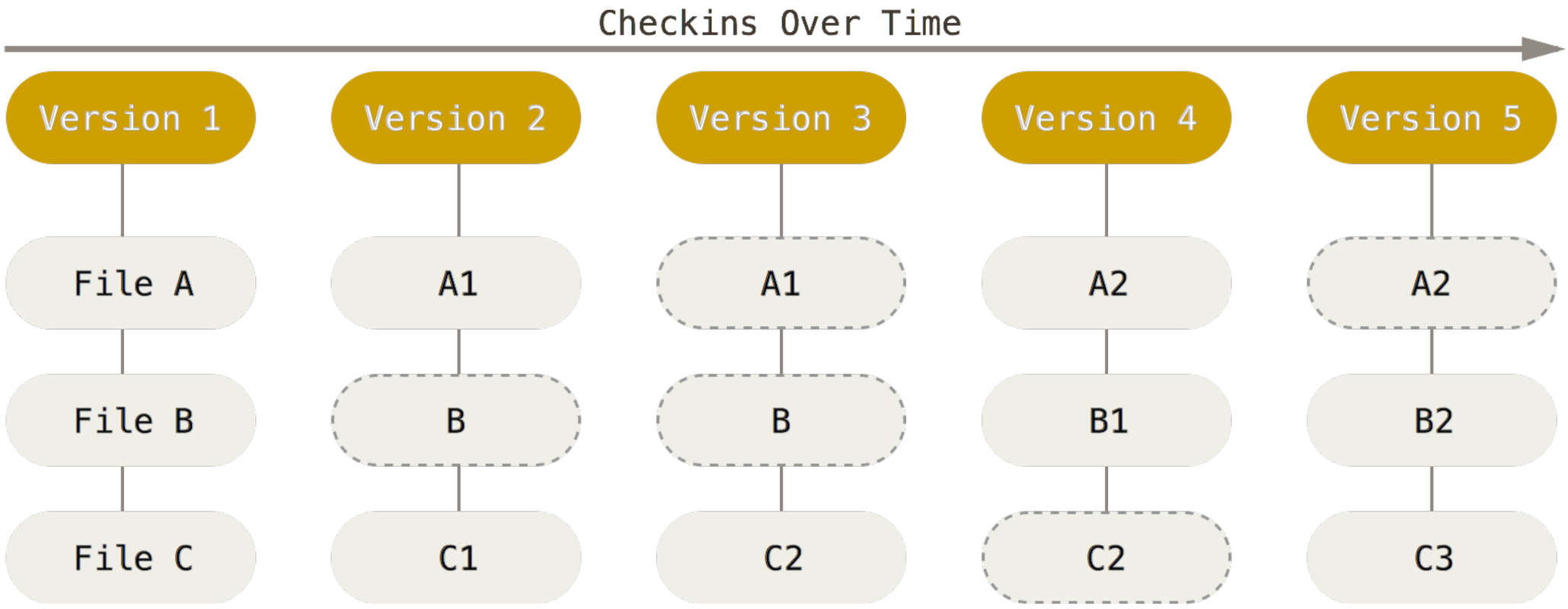
ARTEN VON VCS

- Interne Datenrepräsentation – SVN
  - Änderungen über die Zeit werden gespeichert



ARTEN VON VCS

- Interne Datenrepräsentation – git
  - Daten sind Stream von „Snapshots“



INHALT

- Motivation
- Arten von VCS
- Git Grundlagen
- GitHub
- Git Shell Grundlagen

GIT

- Geschichte
  - Linux Kernel – relativ großes Projekt
  - 1991-2002: kein VCS, Archives + Patches
  - Ab 2002: BitKeeper
  - 2005: Bruch, BitKeeper wird kostenpflichtig
  - Linus Torwards entwickelt Git

DIE 3 HAUPTZUSTÄNDE

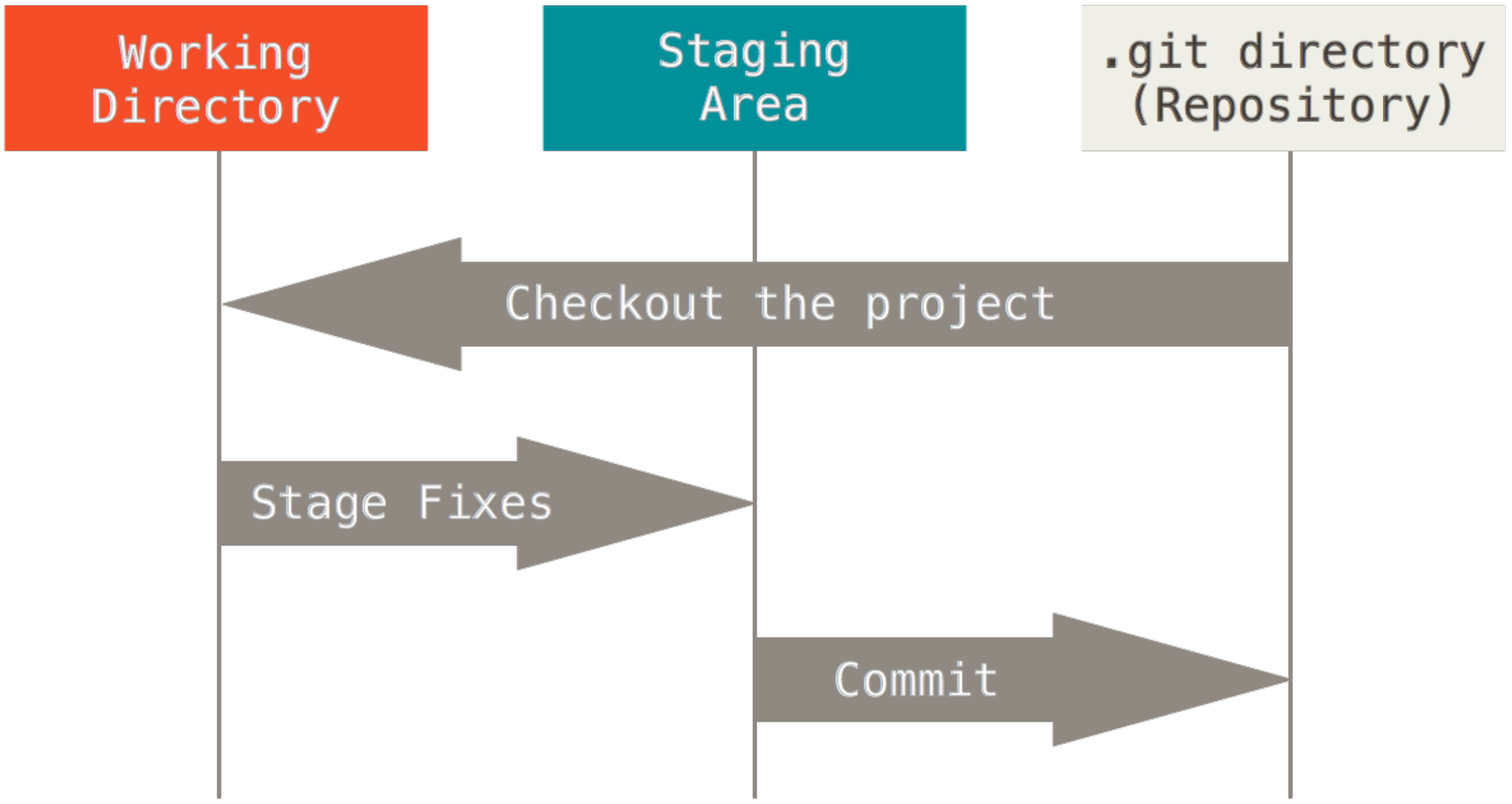
- Git hat 3 Hauptzustände, in denen sich die Files befinden können:
  - **Commited**

    Daten sind sicher in lokaler Datenbank (Repository) gespeichert
  - **Modified**

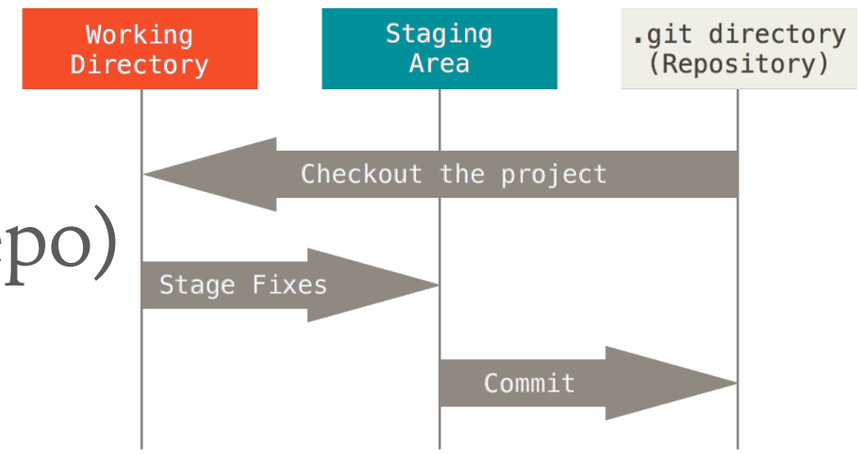
    Daten sind geändert worden, aber noch nicht in Repo gespeichert (commited)
  - **Staged**

    Modifizierte Daten wurden markiert. Beim nächsten commit werden sie in Repo gespeichert

DIE 3 HAUPTZUSTÄNDE



DIE 3 HAUPTZUSTÄNDE



- **.git Ordner:**

  hier speichert Git seine Daten (lokale Repo)

- **Staging Area**

  Datei im .git Ordner, in der gespeichert wird, was beim nächsten commit passieren soll

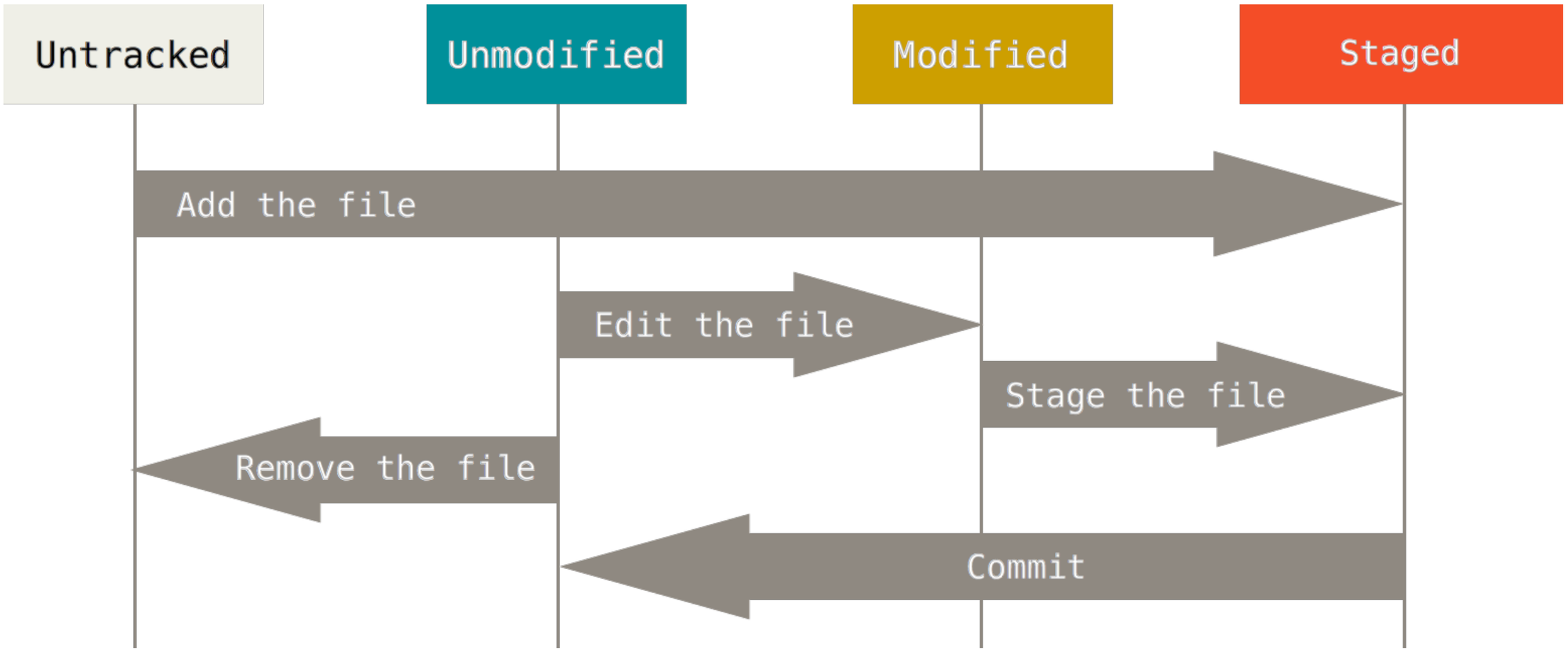
  Wird öfters auch als „Index“ bezeichnet

- **Working Directory:**

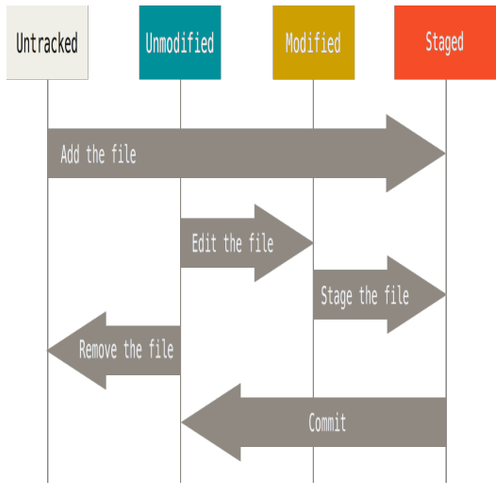
  Checkout von einer Version des Projekts, werden aus Repo herausgezogen (pull)

  Dateien werden hier bearbeitet

ZUSTÄNDE DER FILES



ZUSTAND DER FILES



- **Untracked**

  Dateien, die nicht von git beobachtet werden

- **Unmodified**

  Datei wurde nicht geändert, entspricht also dem ausgecheckten Zustand

- **Modified**

  Datei hat sich geändert

- **Staged**

  Datei wird bei Commit in die Repository kopiert

INHALT

GitHub

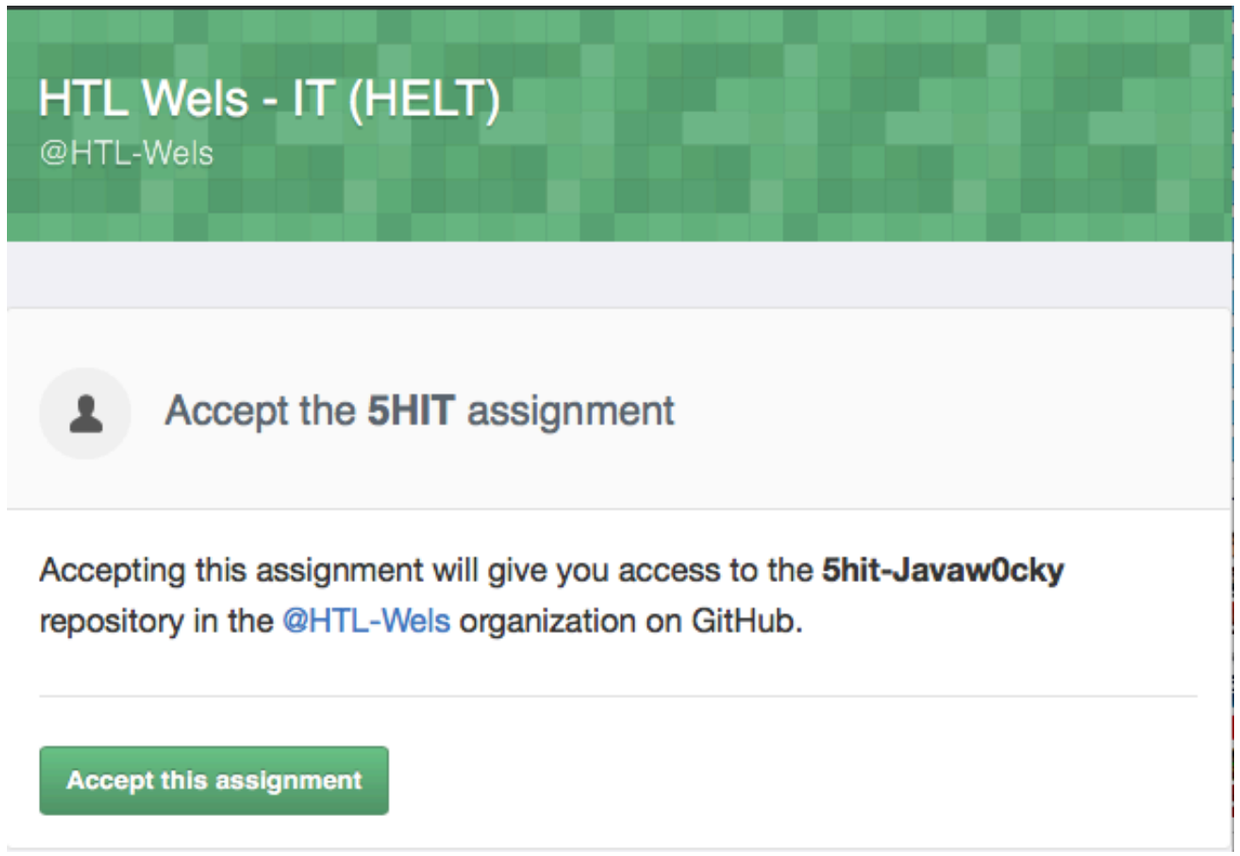
- GitHub Inc. wurde 2007 gegründet
- 2008: Start von GitHub
- > 10 Mio reg. User (08/2015)
- Stellt (public) Git-Repositories für Open Source Entwicklungen gratis zur Verfügung
- Private Repos sind zu bezahlen (ausser für Bildungsbereich)
- 2018 kauft Microsoft GitHub für 6,4 Mrd. Euro

GitHub

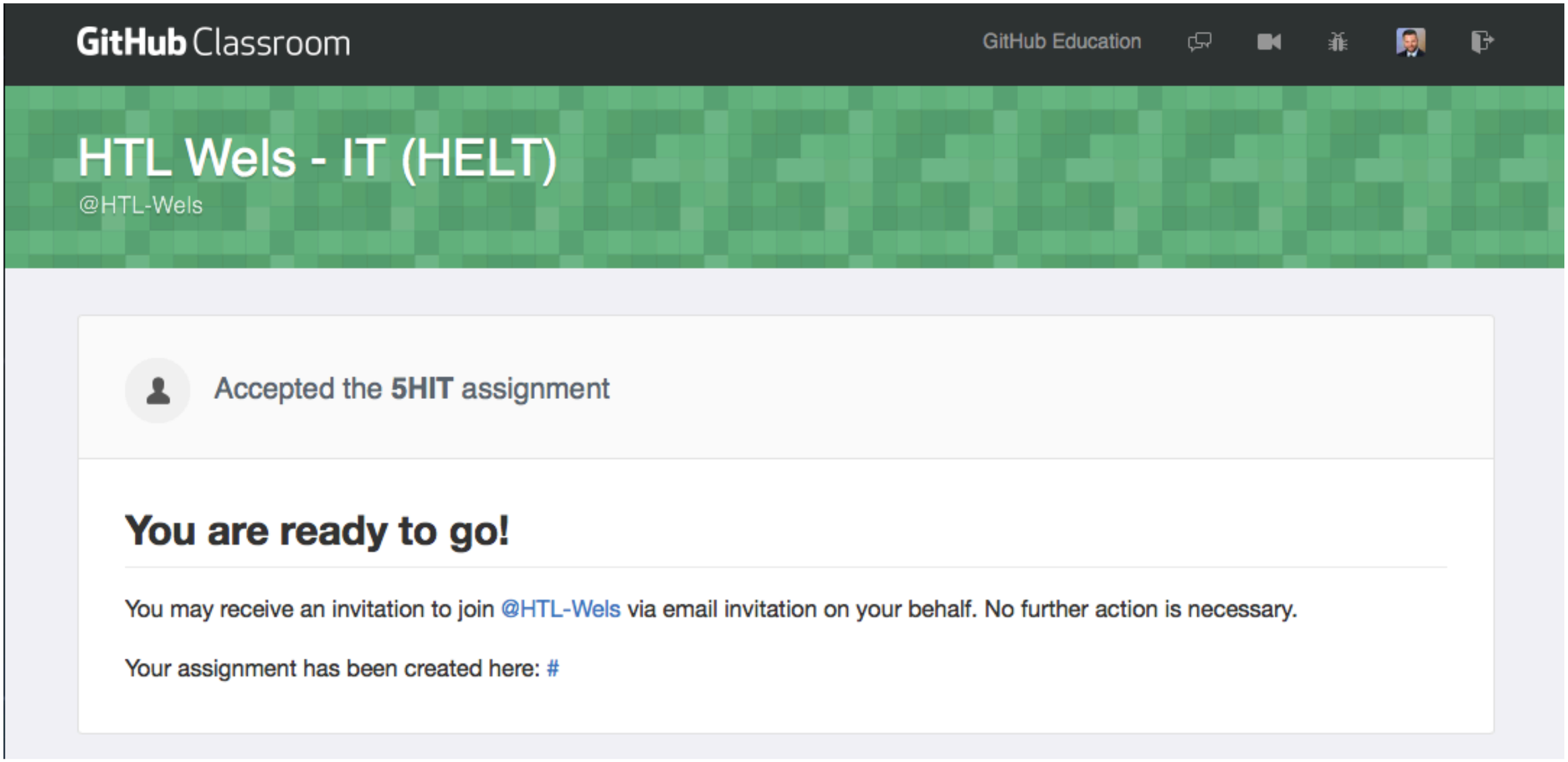
- **Arbeitsauftrag:**
  - Erstelle einen User auf www.github.com
  - Username: NachnameVorname - z.B. ***HelmlThomas***
  - Füge daheim ein Bild von dir im Profil ein
  - Download:
    - GitHub Desktop
    - git for Windows

GITHUB

- https://classroom.github.com/a/NYa_Jjcx



GITHUB


GitHub Classroom
GitHub Education
HTL Wels - IT (HELT)
@HTL-Wels
Accepted the 5HIT assignment
You are ready to go!
You may receive an invitation to join @HTL-Wels via email invitation on your behalf. No further action is necessary.
Your assignment has been created here: #

GITHUB

This repository Search Pull requests Issues Gist

HTL-Wels / 5hit-Javaw0cky-1 Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/HTL-Wels/5hit-Javaw0cky-1.git

We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# 5hit-Javaw0cky-1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/HTL-Wels/5hit-Javaw0cky-1.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/HTL-Wels/5hit-Javaw0cky-1.git
git push -u origin master
```

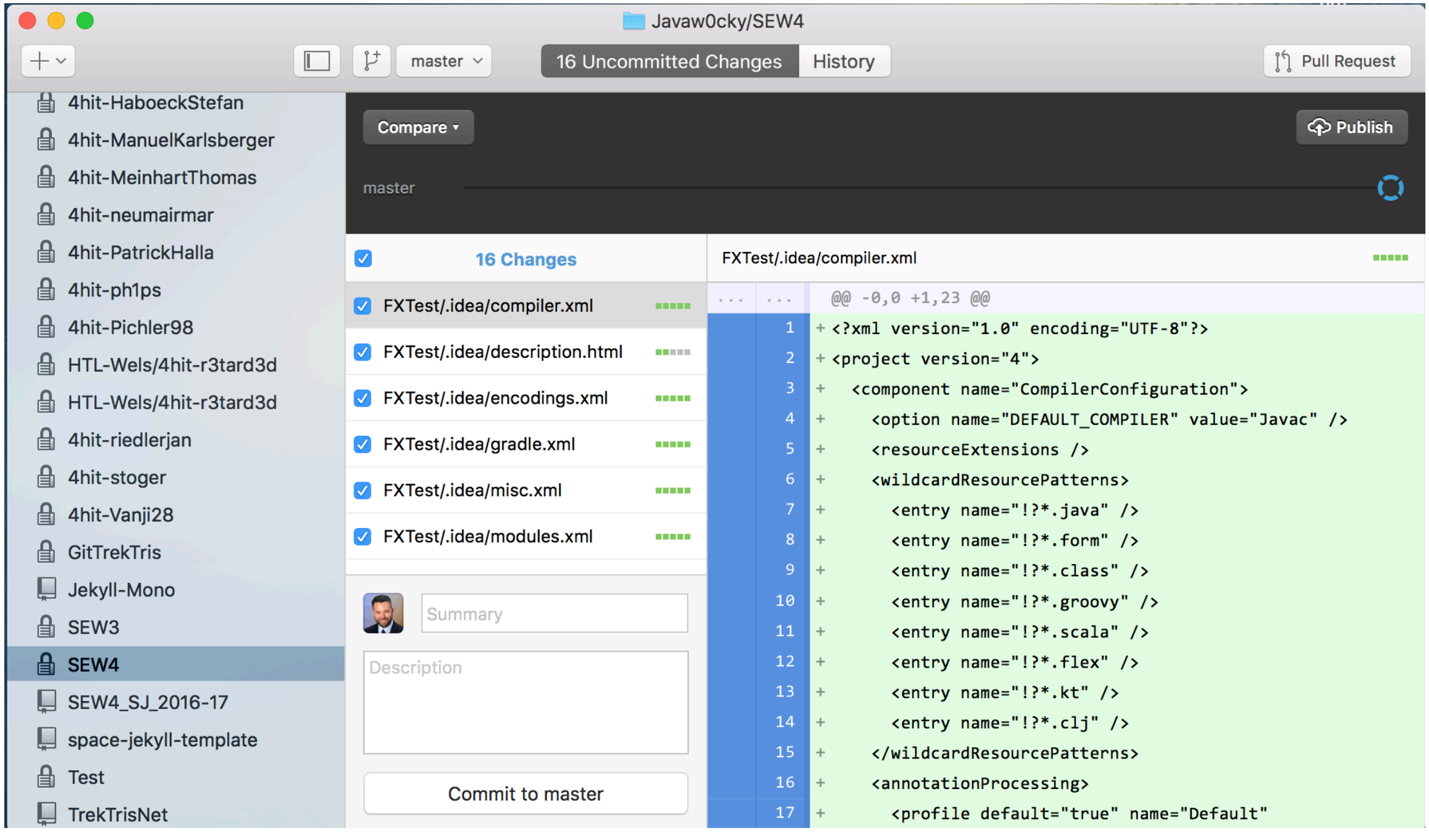
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

GITHUB DESKTOP



GITHUB DESKTOP

- IntelliJ Projekt erstellen in „Git-Projekte-Ordner“
- GitHub Desktop aufrufen
  - Initial commit durchführen
  - Publish
- GitHub Webseite aufrufen
- Class Files, IntelliJ Projektdateien entfernen
- README.md erstellen

INHALT

- Motivation
- Arten von VCS
- Git Grundlagen
- GitHub
- Git Shell Grundlagen

GIT SHELL BEFEHLE

- Erstelle neuen Projekt Ordner / neues IntelliJ Projekt
  - Powershell in GitHub Desktop öffnen
  - Powershell: cd Projekt Ordner

REPOSITORY ERSTELLEN

git init

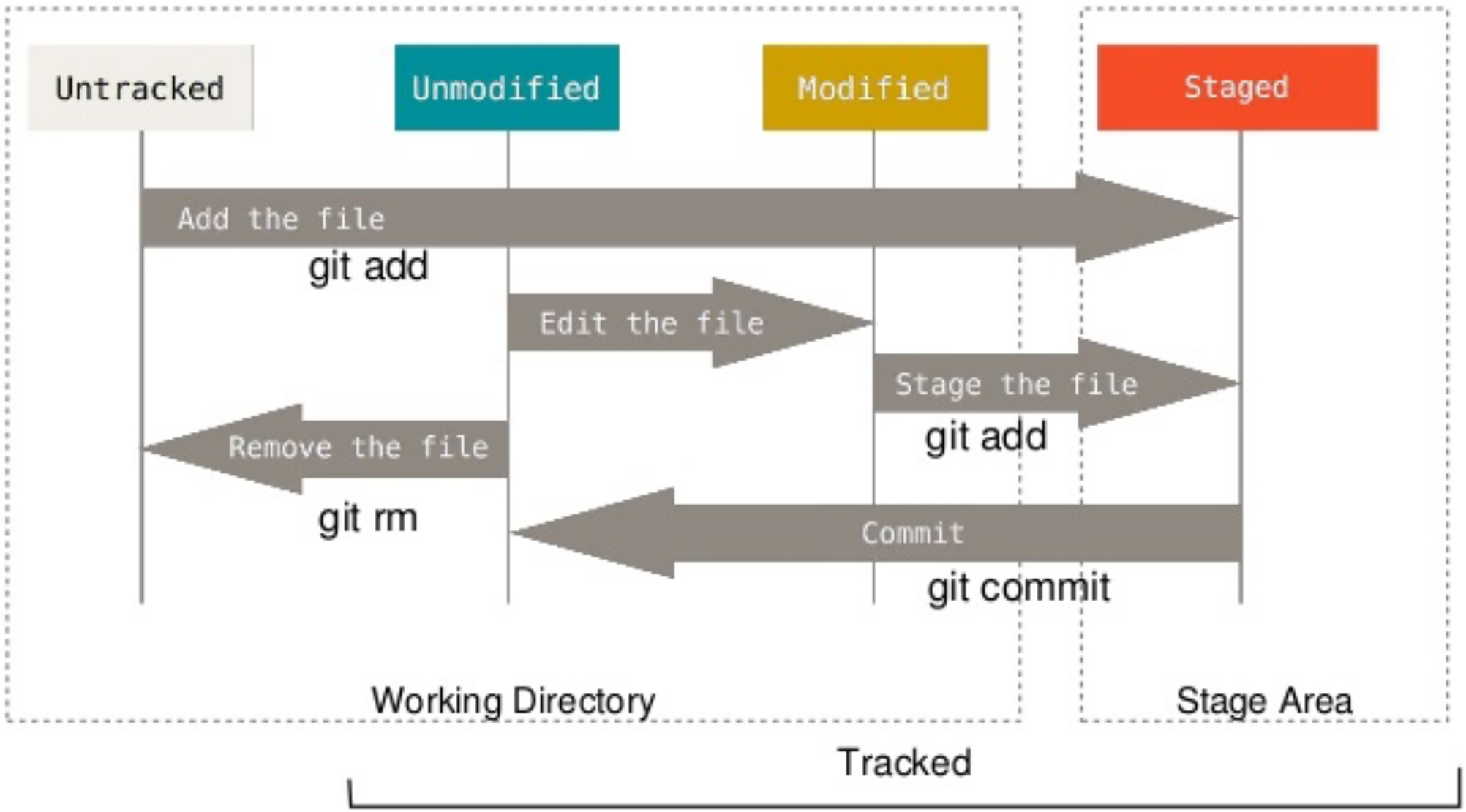
```
BatBook-Pro:IdeaProjects tom$ cd gittest/


BatBook-Pro:gittest tom$ ls -la
total 0
drwxr-xr-x   2 tom  staff    68  5 Feb 13:27 .
drwxr-xr-x  33 tom  staff  1122  5 Feb 13:27 ..

BatBook-Pro:gittest tom$ git init
Initialized empty Git repository in /Users/tom/IdeaProjects/gittest/.git/

BatBook-Pro:gittest tom$ ls -la
total 0
drwxr-xr-x   3 tom  staff   102  5 Feb 13:27 .
drwxr-xr-x  33 tom  staff  1122  5 Feb 13:27 ..
drwxr-xr-x  10 tom  staff   340  5 Feb 13:27 .git
```

STATUS DER GIT REPO

```
git status
```

```
BatBook-Pro:gittest tom$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

FILE STATUS


Untracked
Unmodified
Modified
Staged
Add the file
git add
Edit the file
Stage the file
git add
Remove the file
git rm
Commit
git commit
Working Directory
Stage Area
Tracked

DATEI ERSTELLEN

```
BatBook-Pro:gittest tom$ echo "#Git Test" > README.md


BatBook-Pro:gittest tom$ ls -la
total 8
drwxr-xr-x   4 tom  staff   136  5 Feb 13:34 .
drwxr-xr-x  33 tom  staff  1122  5 Feb 13:27 ..
drwxr-xr-x  10 tom  staff   340  5 Feb 13:29 .git
-rw-r--r--   1 tom  staff    10  5 Feb 13:34 README.md
```

```
BatBook-Pro:gittest tom$ git status
On branch master


Initial commit


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

        README.md

nothing added to commit but untracked files present (use "git add" to track)
```

TRACKING FILES -> STAGING

```
BatBook-Pro:gittest tom$ git add README.md

BatBook-Pro:gittest tom$ git status

On branch master


Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

   new file:   README.md
```

COMMIT

```
BatBook-Pro:gittest tom$ git commit -m "Initial Commit"


[master (root-commit) 6425d14] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

BatBook-Pro:gittest tom$ git status


On branch master

nothing to commit, working tree clean
```

INTELLIJ PROJEKT ERSTELLEN

```
BatBook-Pro:gittest tom$ ls -la
total 16
drwxr-xr-x   8 tom  staff   272  5 Feb 14:02 .
drwxr-xr-x  33 tom  staff  1122  5 Feb 13:27 ..
drwxr-xr-x  13 tom  staff   442  5 Feb 14:02 .git
drwxr-xr-x  10 tom  staff   340  5 Feb 14:02 .idea
-rw-r--r--   1 tom  staff    21  5 Feb 13:45 README.md
-rw-r--r--   1 tom  staff   425  5 Feb 14:02 gittest.iml
drwxr-xr-x   3 tom  staff   102  5 Feb 14:02 out
drwxr-xr-x   3 tom  staff   102  5 Feb 14:02 src
BatBook-Pro:gittest tom$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

          modified:   README.md

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

          .idea/
          gittest.iml
          out/
          src/

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

STAGING MODIFIED FILES

```
BatBook-Pro:gittest tom$ git add .
BatBook-Pro:gittest tom$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   .idea/compiler.xml
    new file:   .idea/description.html
    new file:   .idea/encodings.xml
    new file:   .idea/misc.xml
    new file:   .idea/modules.xml
    new file:   .idea/vcs.xml
    new file:   .idea/workspace.xml
    modified:   README.md
    new file:   gittest.iml
    new file:   out/production/gittest/Main.class
    new file:   src/Main.java
```

UNNÖTIGE FILES ENTFERNEN

```
BatBook-Pro:gittest tom$ git rm -r --cached out/
rm 'out/production/gittest/Main.class'
BatBook-Pro:gittest tom$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .idea/compiler.xml
        new file:   .idea/description.html
        new file:   .idea/encodings.xml
        new file:   .idea/misc.xml
        new file:   .idea/modules.xml
        new file:   .idea/vcs.xml
        new file:   .idea/workspace.xml
        modified:   README.md
        new file:   gittest.iml
        new file:   src/Main.java

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

        out/
```

UNNÖTIGE FILES ENTFERNEN

```
BatBook-Pro:gittest tom$ git rm -r --cached .idea/

rm '.idea/compiler.xml'

rm '.idea/description.html'

rm '.idea/encodings.xml'

rm '.idea/misc.xml'

rm '.idea/modules.xml'

rm '.idea/vcs.xml'

rm '.idea/workspace.xml‘


BatBook-Pro:gittest tom$ git rm --cached gittest.iml

rm 'gittest.iml'
```

UNNÖTIGE FILES ENTFERNEN

```
atBook-Pro:gittest tom$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md
    new file:   src/Main.java

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

    .idea/
    gittest.iml
    out/
```

COMMIT

```
BatBook-Pro:gittest tom$ git commit -m "Hello World hinzugefügt"
[master 903a610] Hello World hinzugefügt
 2 files changed, 7 insertions(+), 1 deletion(-)
 create mode 100644 src/Main.java

BatBook-Pro:gittest tom$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    .idea/
    gittest.iml
    out/

nothing added to commit but untracked files present (use "git add" to track)
```

DATEIEN UMBENENNEN BZW. VERSCHIEBEN

- Werden Dateien verschoben bzw. umbenannt, so bekommt Git das nicht korrekt mit
- Daher gibt es ein extra Kommando dafür: mv

```
BatBook-Pro:gittest tom$ git mv README.md readme.md
BatBook-Pro:gittest tom$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    renamed:    README.md -> readme.md
```

DATEIEN IGNORIEREN

- in der Datei .gitignore können Dateien angegeben werden, die automatisch von git ignoriert werden:

```
# class files
*.class
# Package Files #
*.jar*.war*.ear
```

GIT LOG

```
BatBook-Pro:gittest tom$ git log
commit 903a610db0d57a89c877b5b95785d8a31134399c
Author: Thomas Helml <thomas.helml@htl-wels.at>
Date:   Sun Feb 5 14:14:40 2017 +0100

    Hello World hinzugefügt

commit 6425d1425032787303 78e1d446c98a1e1cf799a9
Author: Thomas Helml <thomas.helml@htl-wels.at>
Date:   Sun Feb 5 13:39:07 2017 +0100

    Initial Commit
```

ÄNDERUNGEN RÜCKGÄNGIG

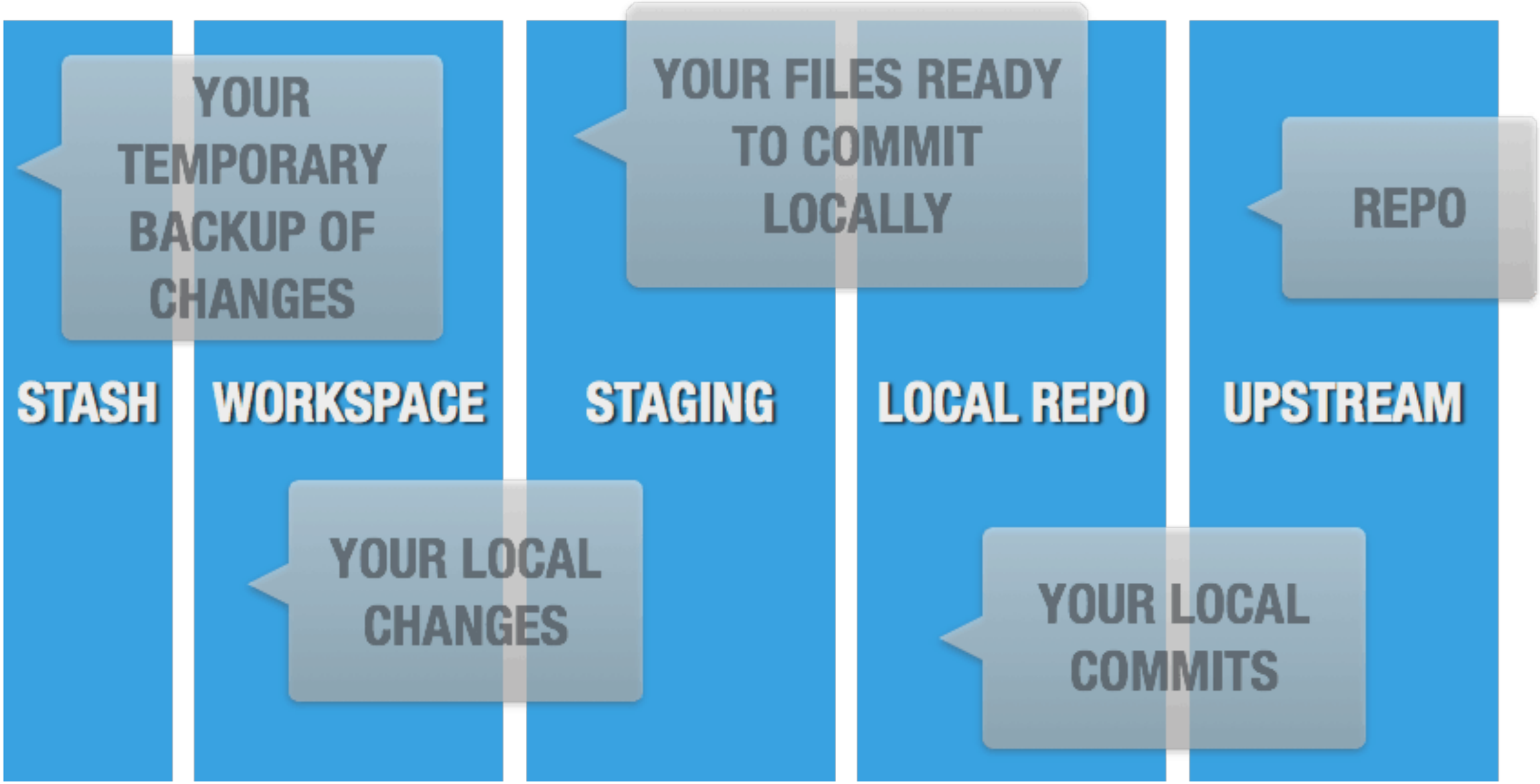
```
BatBook-Pro:gittest tom$ echo "neuer inhalt" > readme.md


BatBook-Pro:gittest tom$ cat readme.md
neuer inhalt

BatBook-Pro:gittest tom$ git checkout readme.md
BatBook-Pro:gittest tom$ cat readme.md
#Git Test - modified
```

AUF ALTEN COMMIT RÜCKGÄNGIG

```
BatBook-Pro:gittest tom$ git log
commit 903a610db0d57a89c877b5b95785d8a31134399c
Author: Thomas Helml <thomas.helml@htl-wels.at>
Date:   Sun Feb 5 14:14:40 2017 +0100

    Hello World hinzugefügt

commit 6425d142503278730378e1d446c98a1e1cf799a9
Author: Thomas Helml <thomas.helml@htl-wels.at>
Date:   Sun Feb 5 13:39:07 2017 +0100

    Initial Commit
```

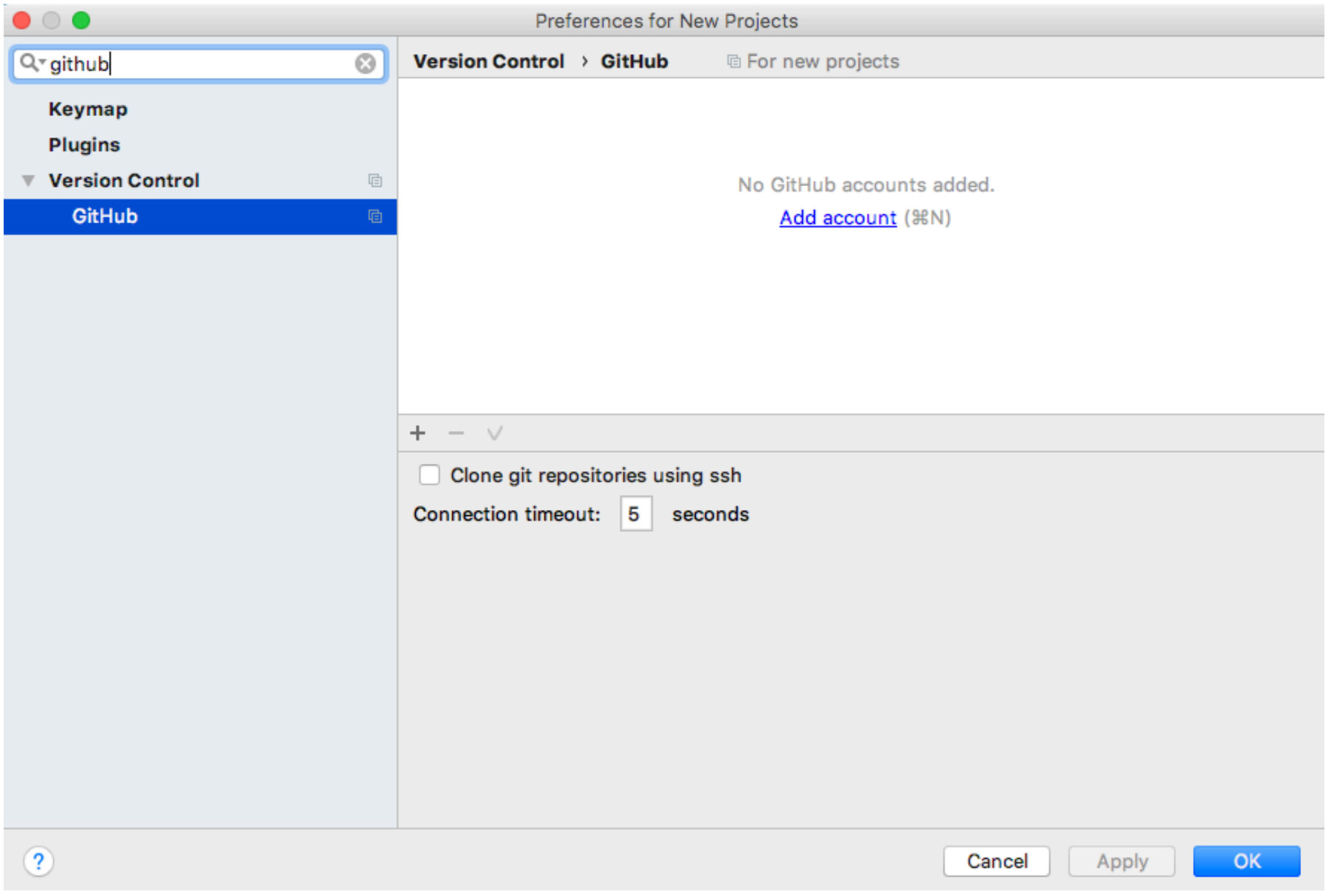
```
git checkout 6425d142503278730378e1d446c98a1e1cf799a9
Previous HEAD position was 903a610... Hello World hinzugefügt
HEAD is now at 6425d14... Initial Commit
```

DIE AREAS

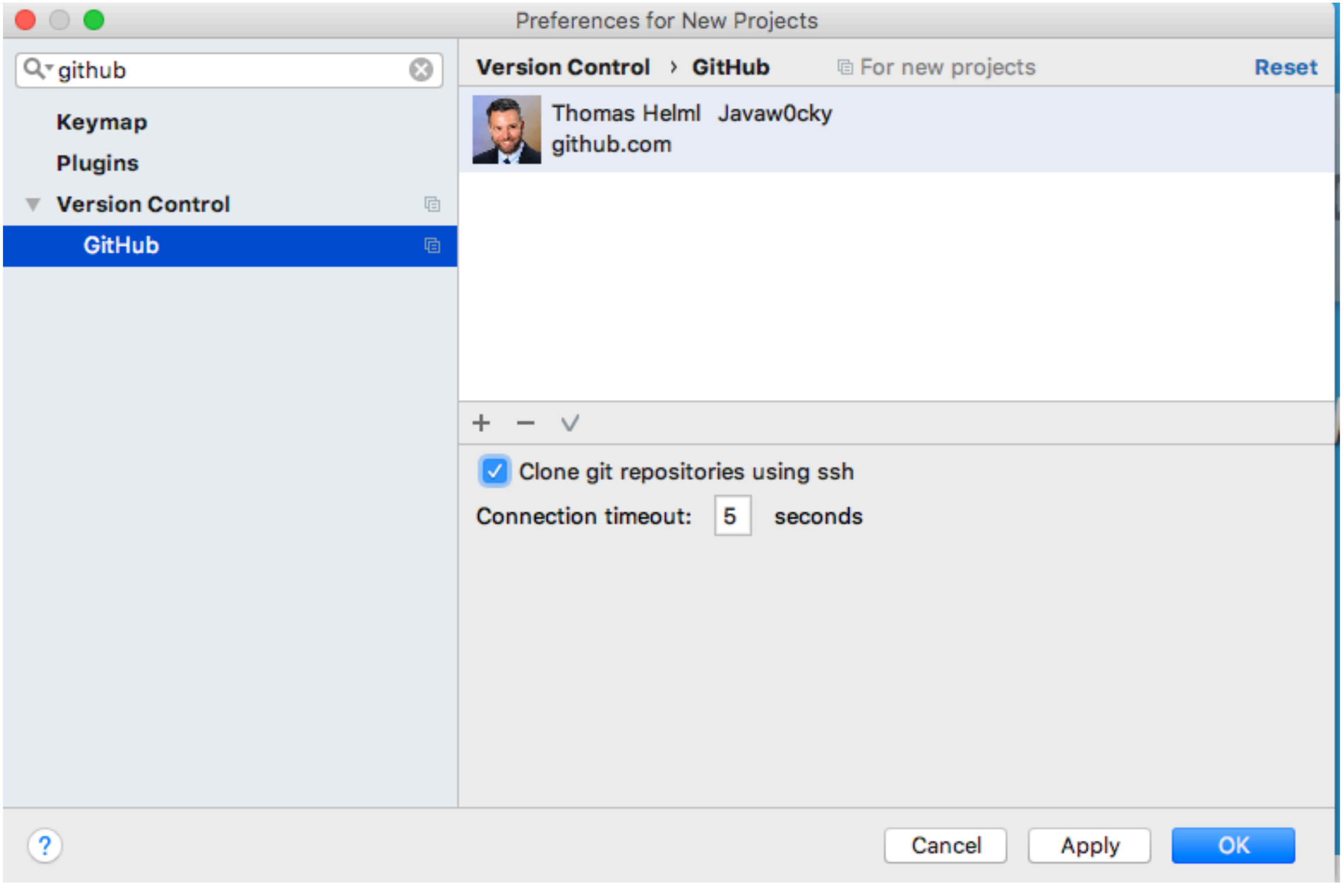

YOUR TEMPORARY BACKUP OF CHANGES
YOUR FILES READY TO COMMIT LOCALLY
REPO
STASH
WORKSPACE
STAGING
LOCAL REPO
UPSTREAM
YOUR LOCAL CHANGES
YOUR LOCAL COMMITS

GITHUB IN INTELLIJ

- Arbeitsauftrag:
  - Richte GitHub in IntelliJ ein:



GITHUB IN INTELLIJ



QUELLEN

- http://www.slideshare.net/paradigmatecnologico/git-vs-subversion
- https://github.com/CourseReps/ECEN489-Fall2015/wiki/git
- https://git-scm.com/book/en/v2/Getting-Started-Git-Basics