

Eine kurze Einführung in **git**

Versionsverwaltung

Ausgangslage

- Softwareentwicklung ist Teamarbeit
- Software besteht aus vielen Dokumenten

Versionsverwaltung

Konsequenz

- Verschiedene Personen greifen (gleichzeitig) auf Dokumente zu
- Oft bearbeiten verschiedene Personen gleichzeitig (unabhängig voneinander) dasselbe Dokument

Versionsverwaltung

Typische Probleme/ Fragen

- Wo/was ist die aktuelle Version?
- Was ist die zuletzt lauffähige Version?
- Welche Version wurde dem Kunden „Schäfer“ präsentiert?

Versionsverwaltung

Einfache Lösungen

- Austausch der Dokumente via USB-Stick/ Festplatte, Netzwerkfestplatte
- Austausch der Dokumente via Mail
- Austausch via Cloud

Versionsverwaltung

Einfache Lösungen erzeugen neue Probleme

- Koordination
- Manuelle Verwaltung
- Versions- und Änderungsfragen nicht bzw. nur schwer beantwortbar

Versionsverwaltung

„Sinnvolle“ Lösung

- Versions- und Konfigurationsmanagementsystem
- Lösen alle genannten Probleme (fast) ohne Zusatzaufwand
- Bieten sogar Zusatzleistungen (z.B. einfache Datensicherung)

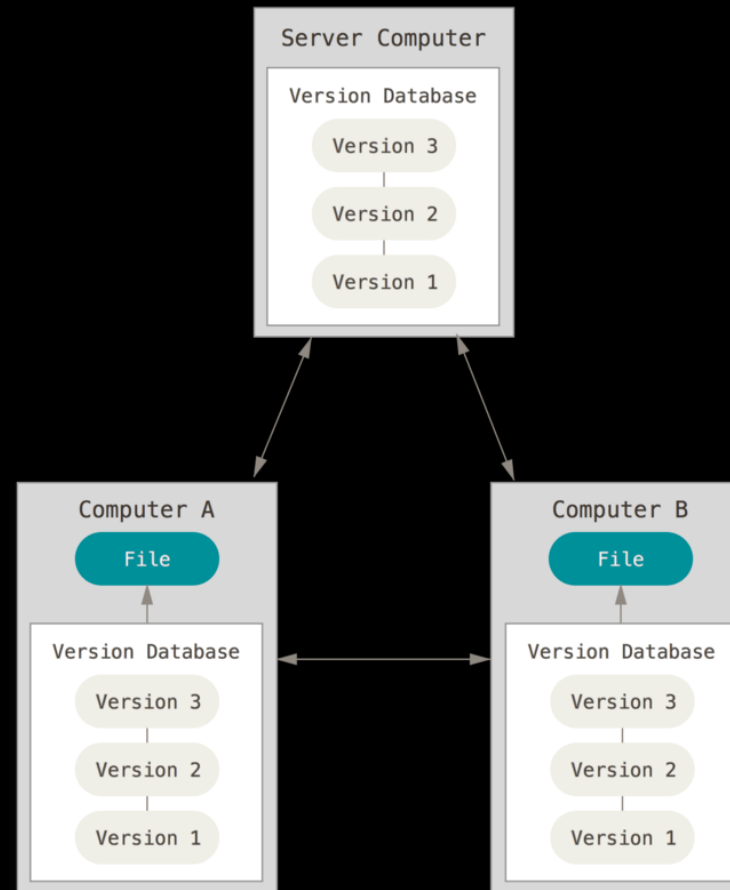
git

- git (engl. Slang für Blödmann)

“The joke ‘I name all my projects for myself, first Linux, then git’ was just too good to pass up. But it is also short, easy-to-say, and type on a standard keyboard. And reasonably unique and not any standard command, which is unusual.”

Was ist git?

- (Verteilte) Versionsverwaltung

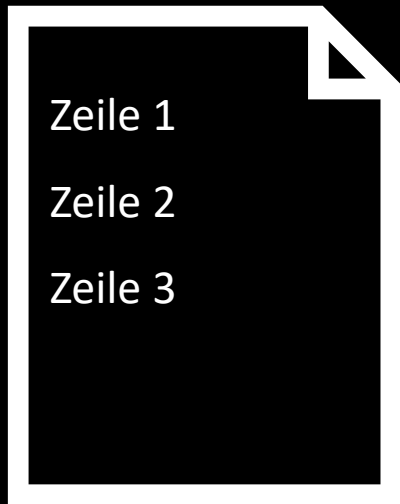


Quelle: [3]

Anwendungsgebiete: Allgemein

- Austausch von Dateien mit anderen
- Historie, die nachvollziehbar ist
- Remote Backup

Änderungen in Dateien verfolgen



Erstelle ein File



Füge eine Zeile hinzu



Entferne eine Zeile

Anwendungsgebiete: Programmieren

- Code zwischen verschiedenen Menschen synchronisieren
- Codeänderungen können getestet werden, ohne das Original zu verändern
- Zurückkehren zu alten Versionen

Schlüsselkonzepte von git

- nicht-lineare Entwicklung (branch)
- Vollständiges Speichern von Daten
- Dezentralität (clone)

git clone

- Erstellt eine Kopie eines repository
- Speichert diese auf ihrem Computer
- Ein „fork“ erstellt eine eigene Kopie von einem repository eines anderen Benutzers

git checkout

- Auswahl einer branch

git commit

- speichert die Änderungen im repository als eine neue Version („commit“)
- protokolliert eine Nachricht

git push

- sendet übertragene Änderungen an das remote repository

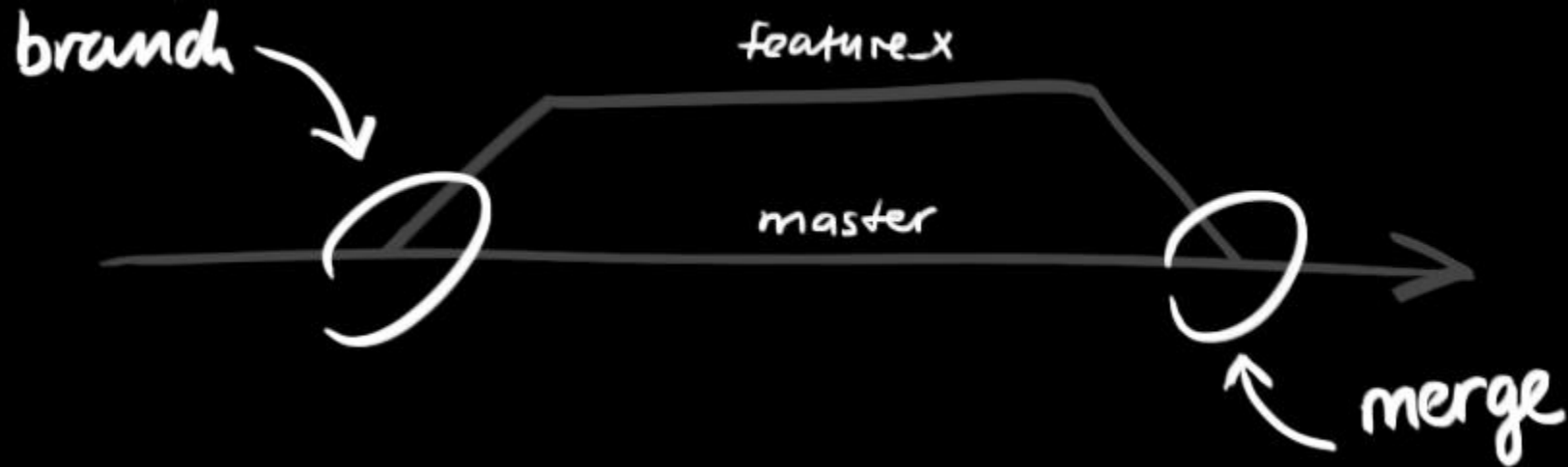
git pull

- lädt Änderungen aus dem remote repository

Branching (Entwicklungslinien)

- Eine branch ist eine Version des repositorys
- Jede branch hat eine eigene commit history und eine aktuelle Version

Branching (Entwicklungslinien)



Quelle: [1]

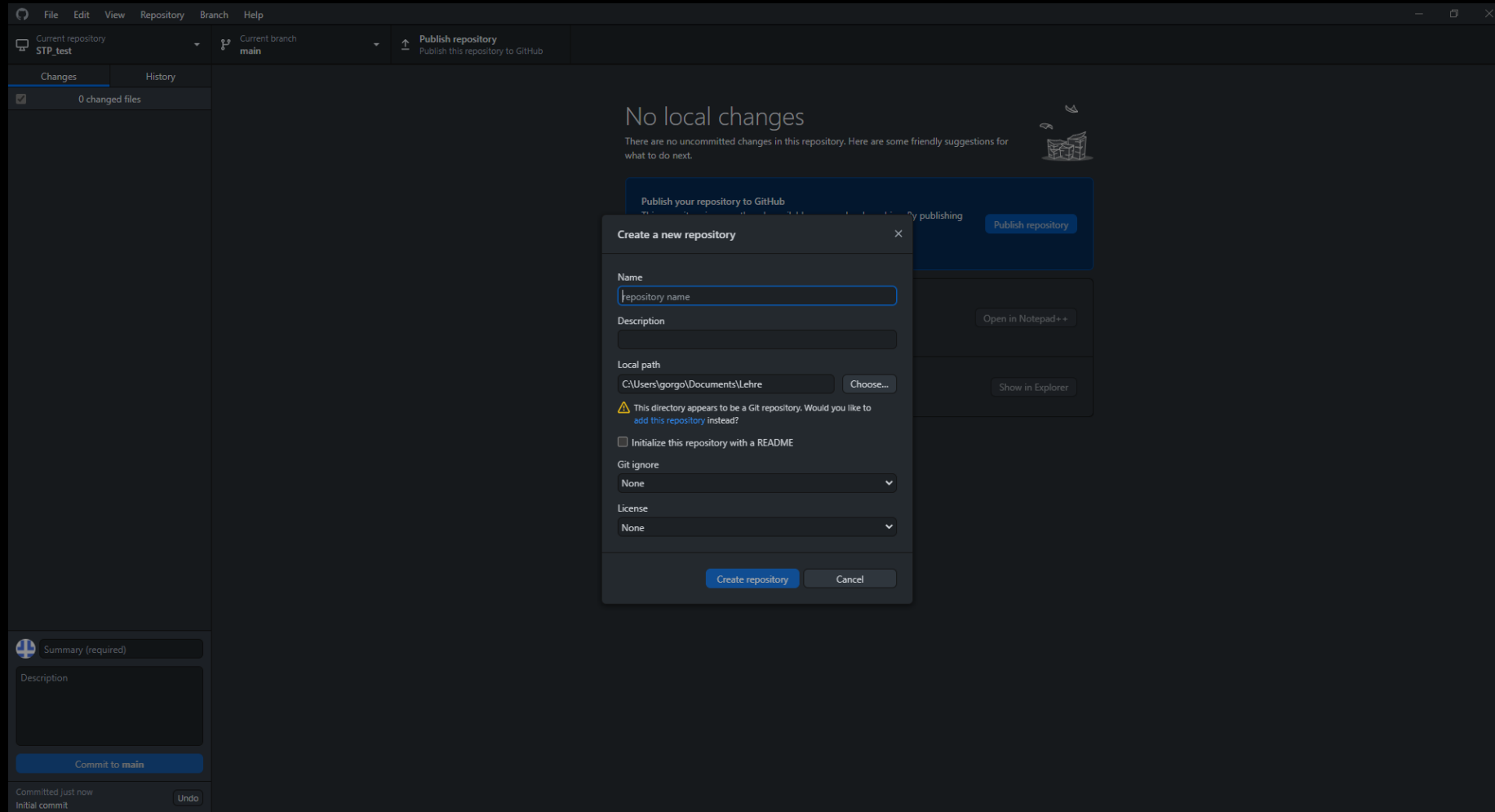
Merge

- Zusammenführen von branches

Merge-Konflikte

- Treten auf, wenn verschiedene commits nicht automatisch zusammengeführt werden können
- müssen gelöst werden

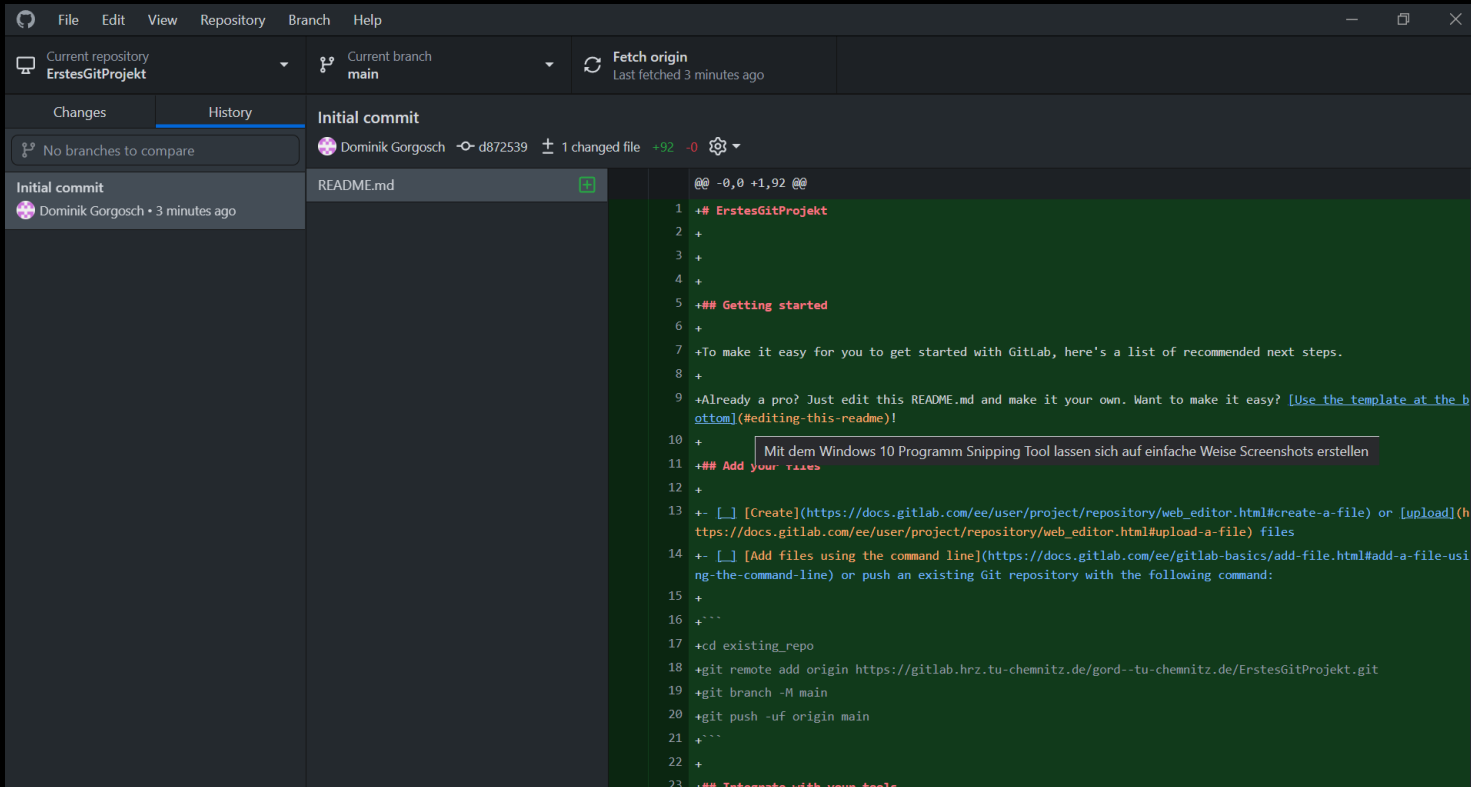
Beispiel: Erstellen eines neuen Repositories



Beispiel: Klonen eines „leeren“ Repositories

d872539

1



The screenshot shows the GitLab web interface for a repository named 'ErstesGitProjekt'. The user is logged in as Dominik Gorgosch. The interface displays the 'Initial commit' page, showing a single file 'README.md' with 1 changed file (+92 -0). The commit message is 'Initial commit'. The README.md content is displayed in a code editor, showing a template for a README file with sections for 'Getting started', 'Add your team', and 'Integrate with your tools'. The code is as follows:

```
@@ -0,0 +1,92 @@
1  +# ErstesGitProjekt
2  +
3  +
4  +
5  +## Getting started
6  +
7  +To make it easy for you to get started with GitLab, here's a list of recommended next steps.
8  +
9  +Already a pro? Just edit this README.md and make it your own. Want to make it easy? [Use the template at the bottom](#editing-this-readme)!
10 +
11 +  Mit dem Windows 10 Programm Snipping Tool lassen sich auf einfache Weise Screenshots erstellen
12 +
13 +  [ ] [Create](https://docs.gitlab.com/ee/user/project/repository/web_editor.html#create-a-file) or [upload](https://docs.gitlab.com/ee/user/project/repository/web_editor.html#upload-a-file) files
14 +  [ ] [Add files using the command line](https://docs.gitlab.com/ee/gitlab-basics/add-file.html#add-a-file-using-the-command-line) or push an existing Git repository with the following command:
15 +
16 +  ```
17 +  cd existing_repo
18 +  git remote add origin https://gitlab.hrztu-chemnitz.de/gord--tu-chemnitz.de/ErstesGitProjekt.git
19 +  git branch -M main
20 +  git push -uf origin main
21 +  ```
22 +
23 +  ## Integrate with your tools
```



README.md

Workspace



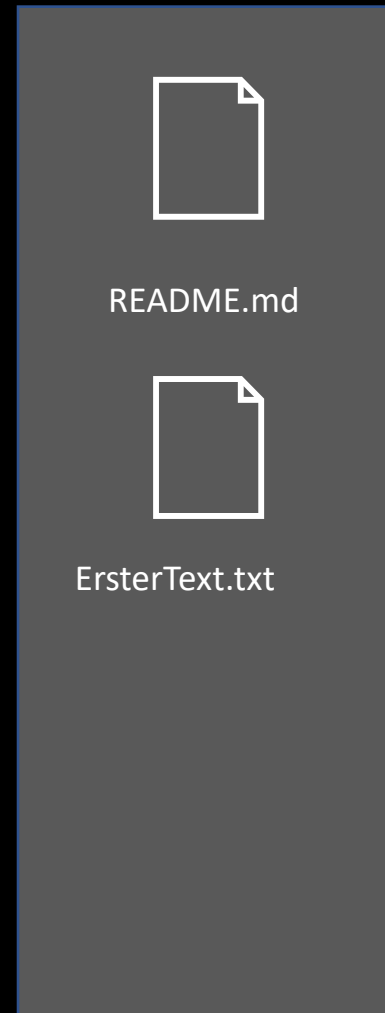
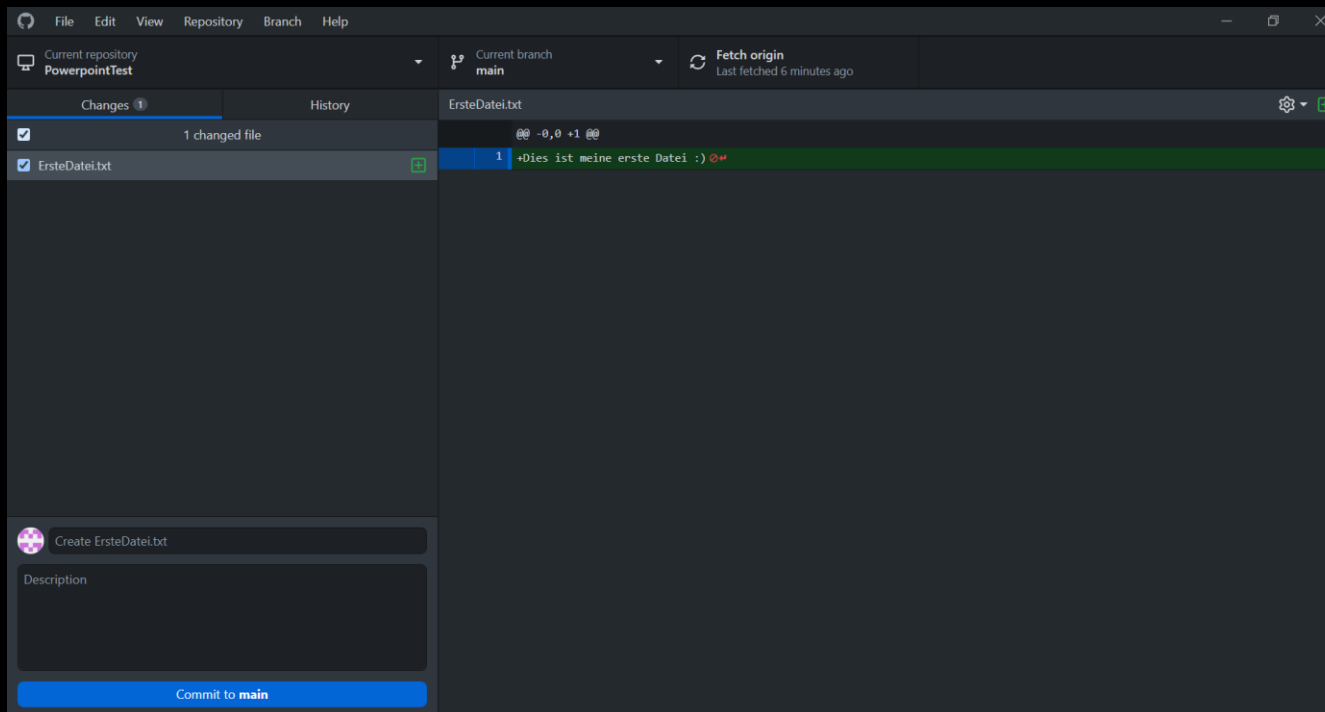
README.md

Repository

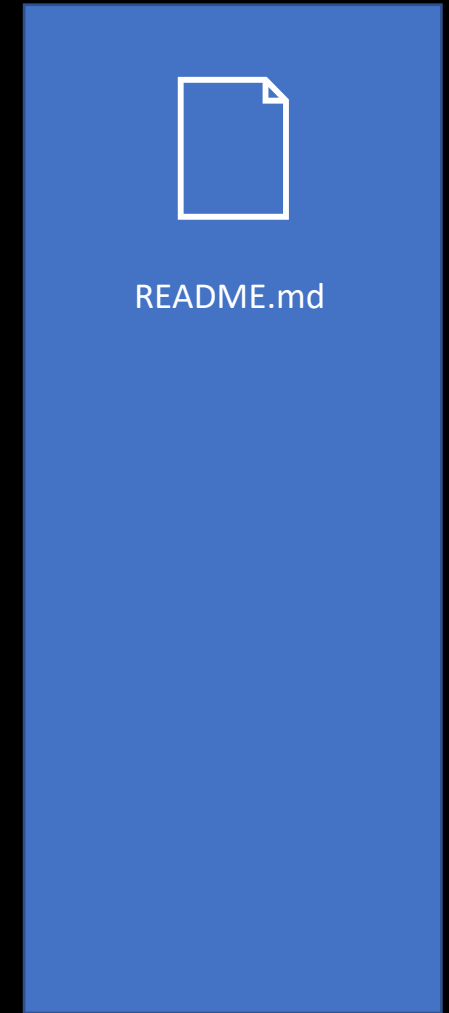
Beispiel: Lokal eine Datei dem Repository hinzufügen

d872539

1



Workspace



Repository

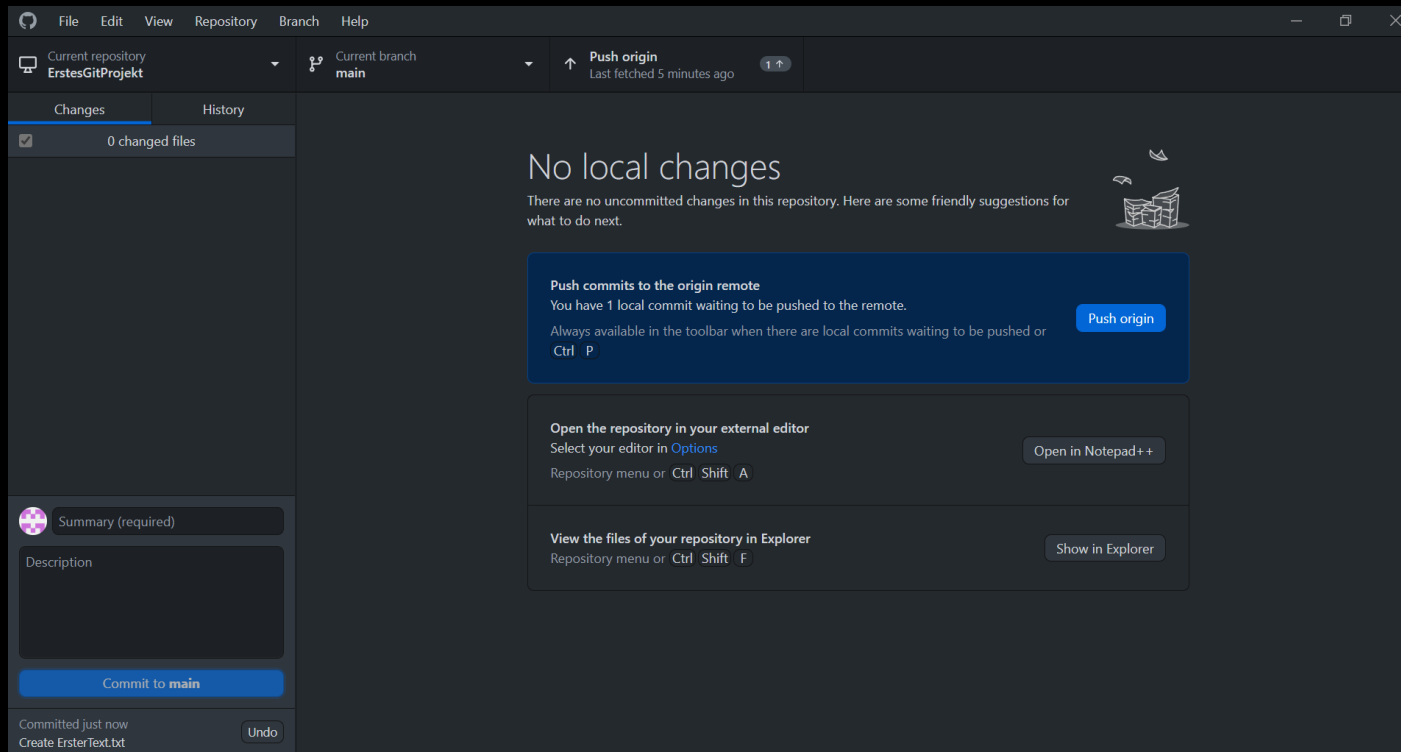
Beispiel: Lokale Änderungen committen

d872539

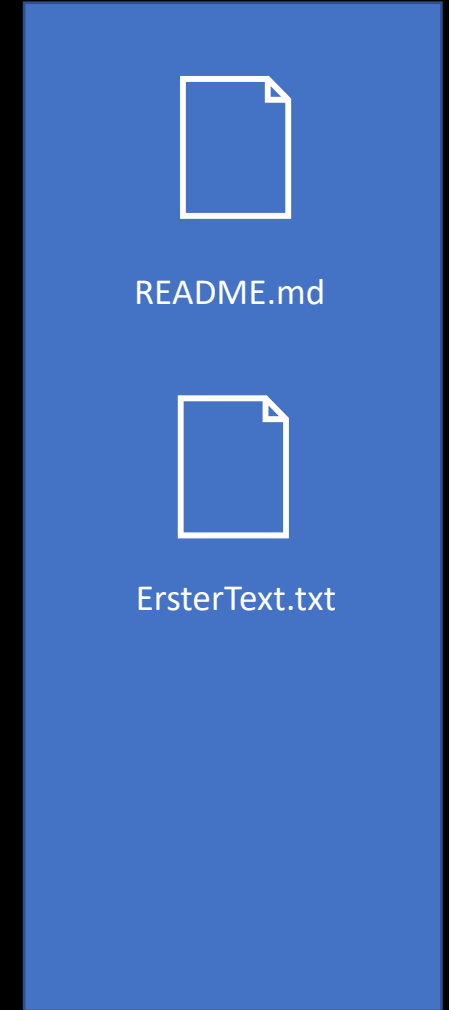
40771f7

1

2



Workspace



Repository

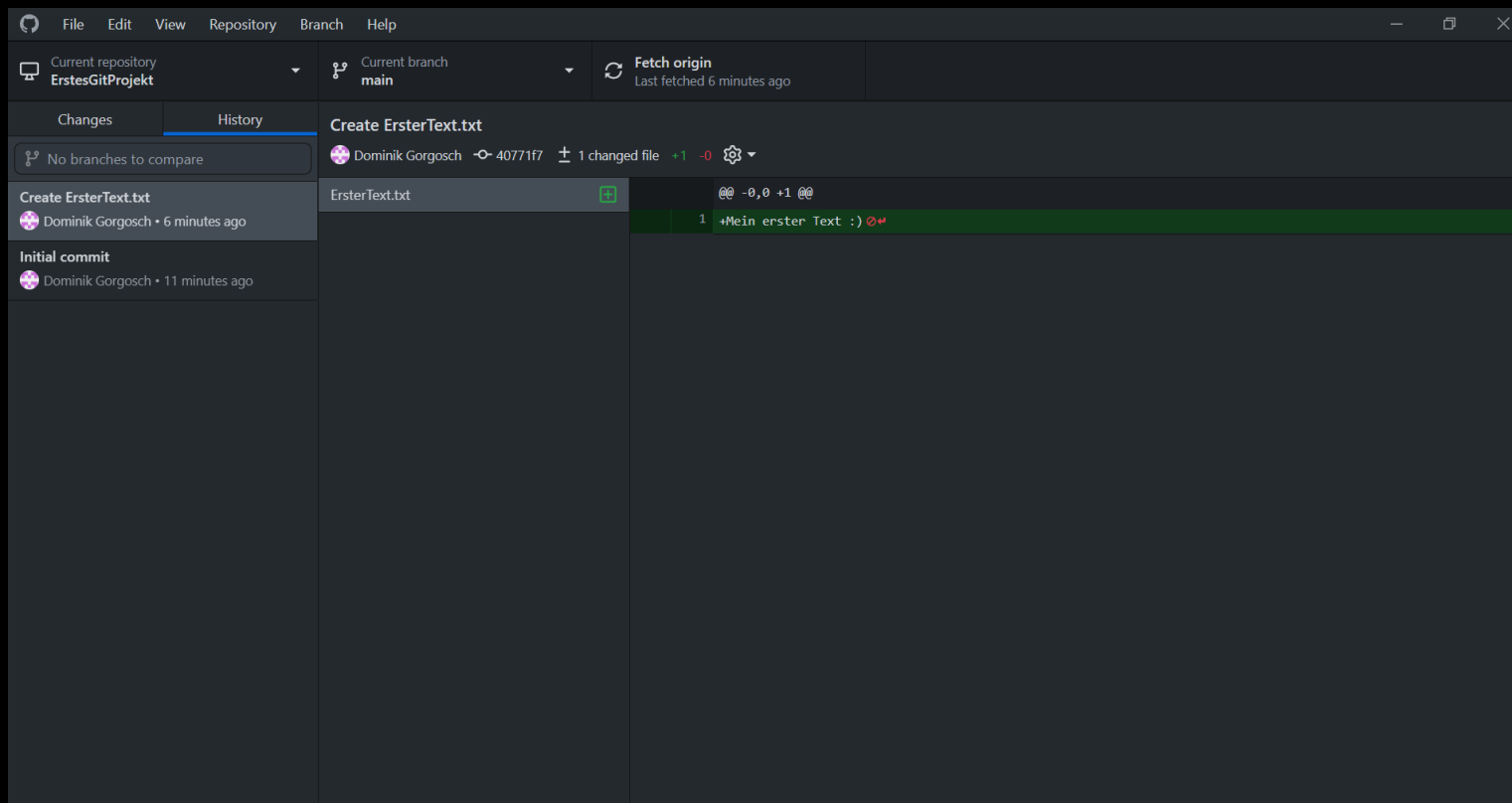
Beispiel: Änderungen pushen

d872539

40771f7

1

2



README.md



ErsterText.txt

Workspace



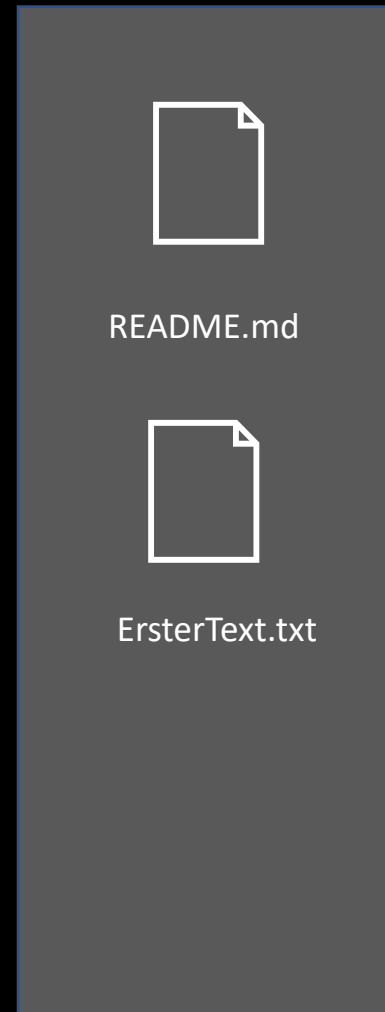
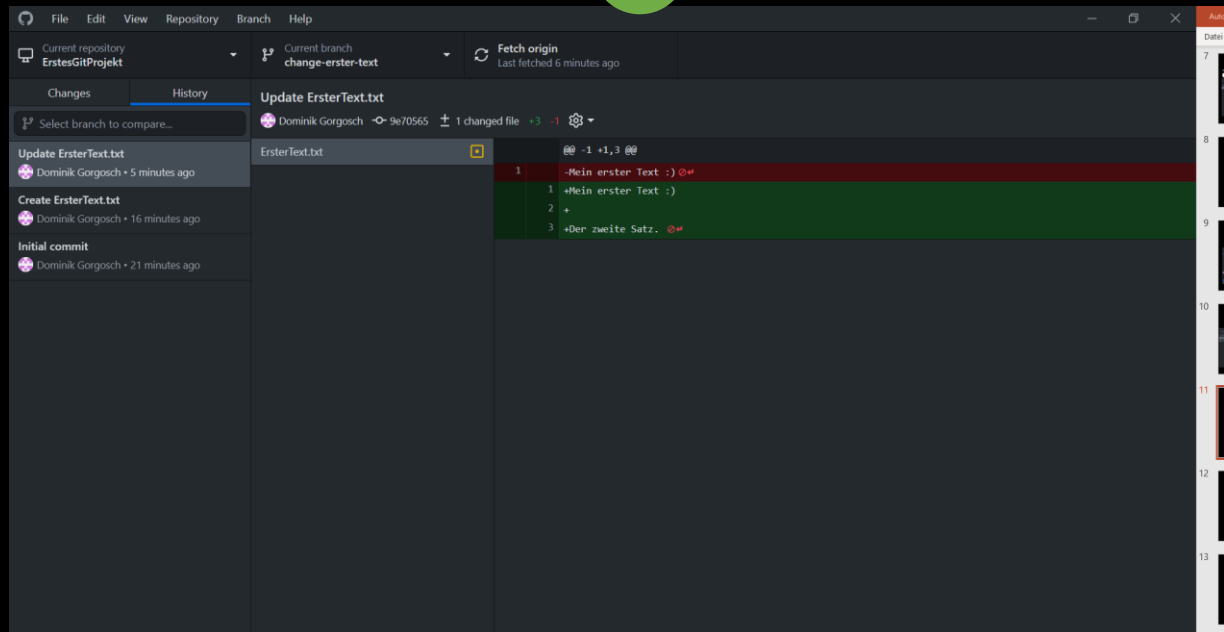
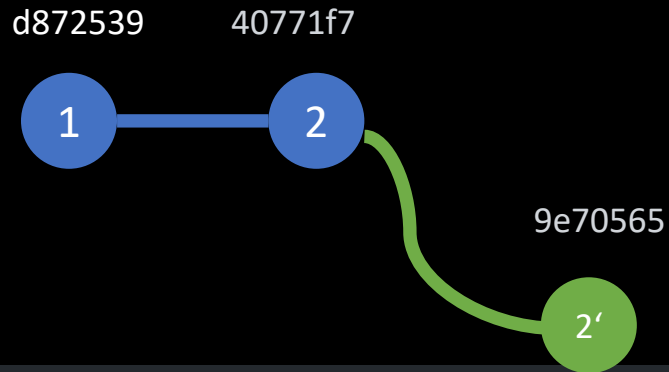
README.md



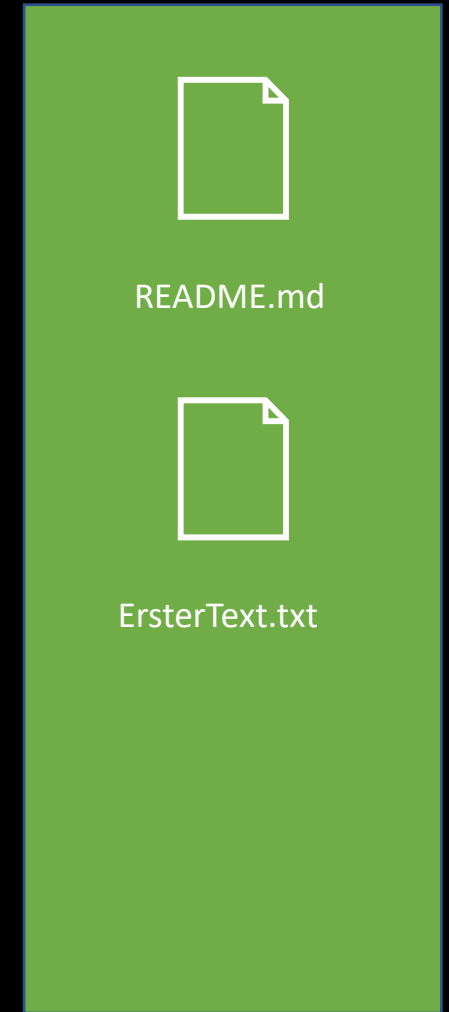
ErsterText.txt

Repository

Beispiel: Erstellen einer neuen Branch

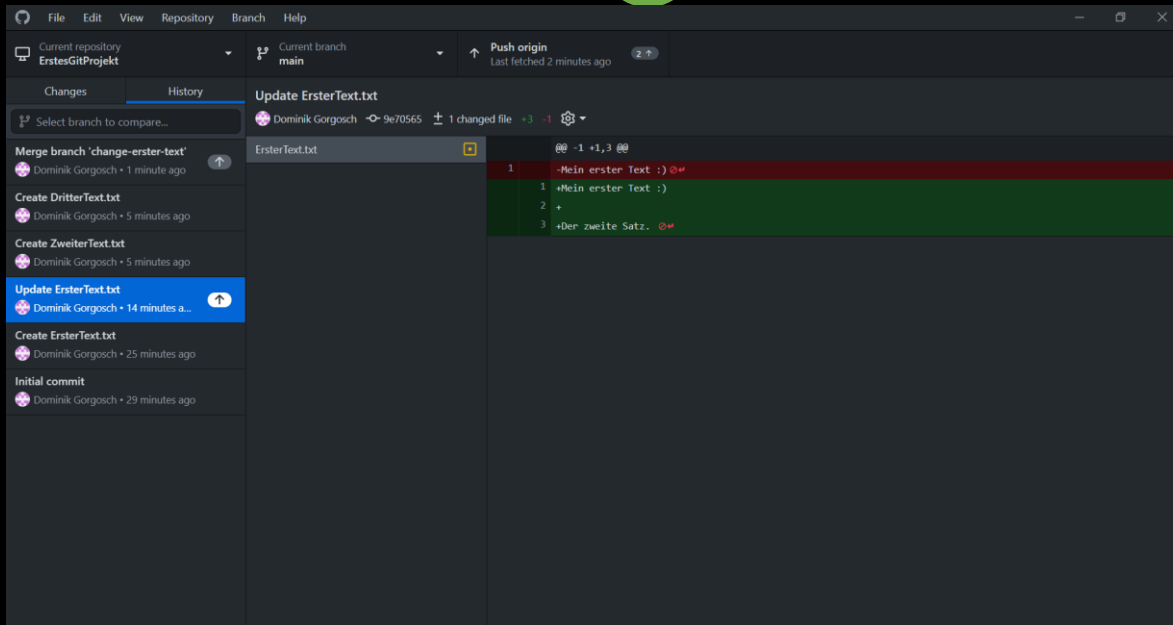
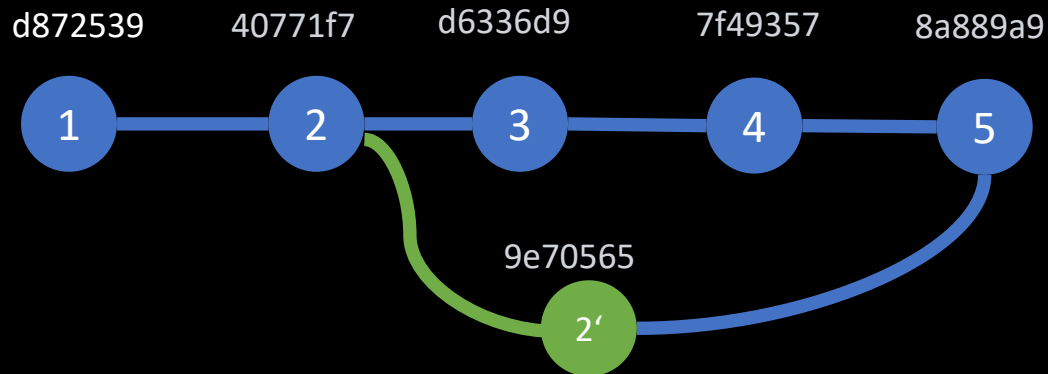


Workspace

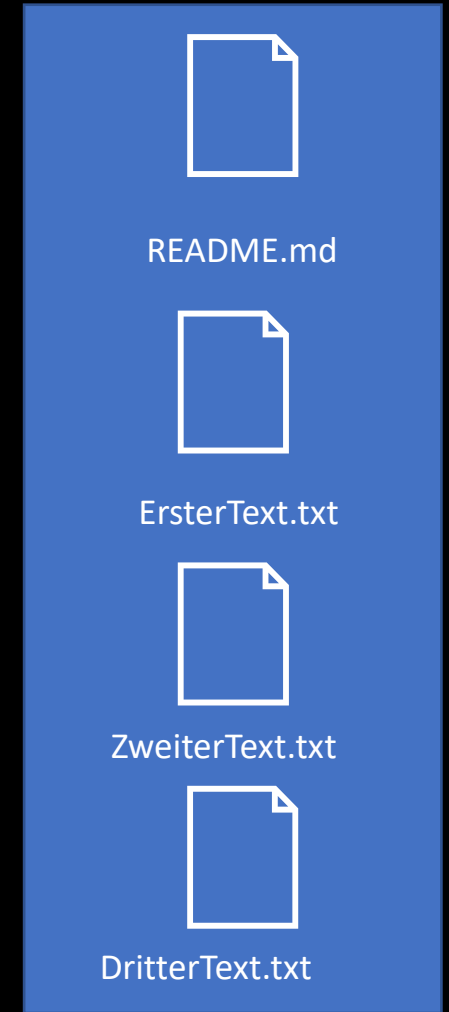


Repository

Beispiel: Branches mergen



Workspace



Repository

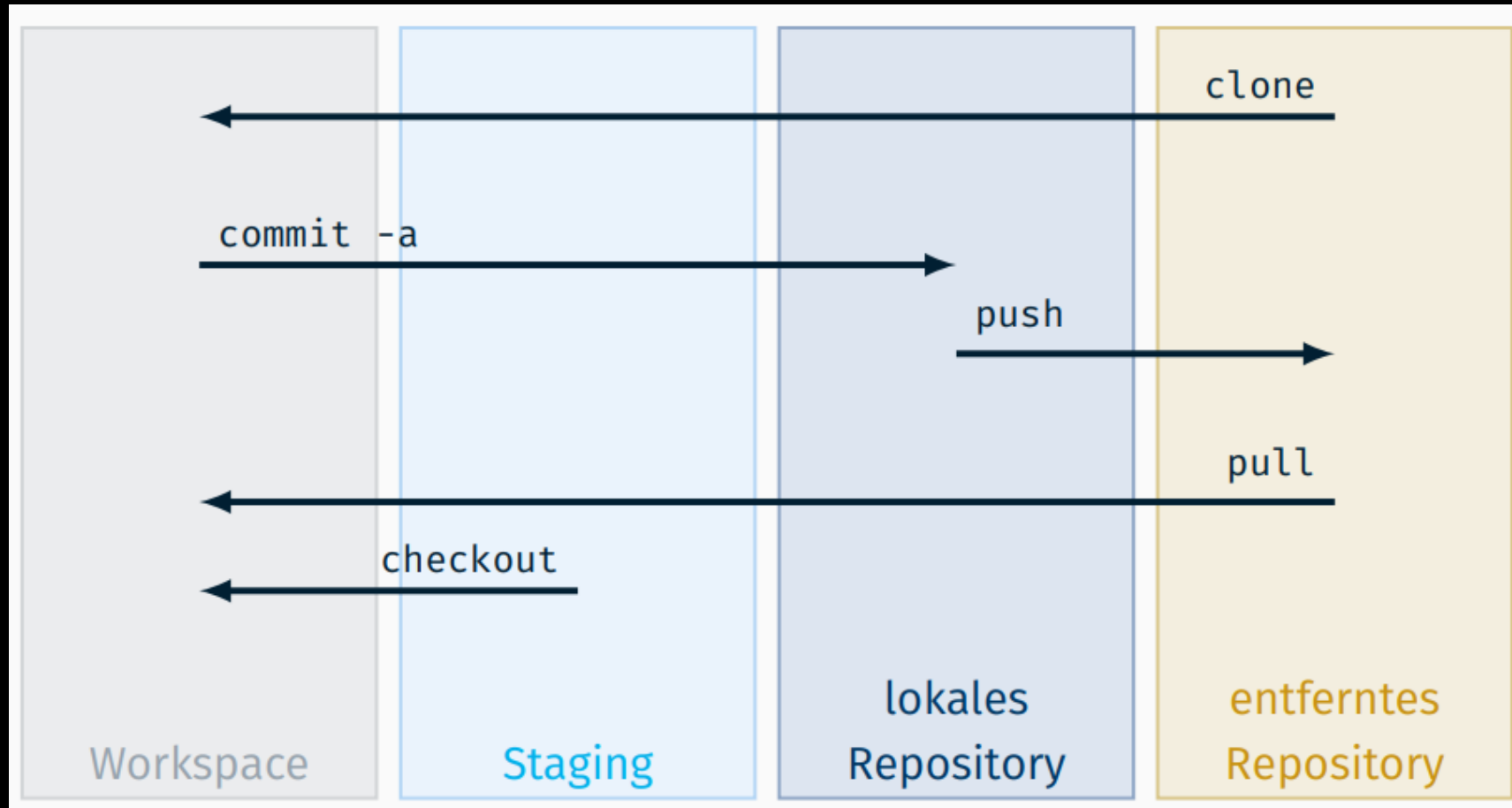
Inhalt für das Repository

Welche Dokumente und Dateien gehören ins Repository?

Alle Dokumente und Dateien, ihrer Software und ihrem Entwicklungsprozess

Vermeiden Sie Kopien anzulegen!

Überblick der verschiedenen Befehle



Quellen

[1] <https://rogerdudler.github.io/git-guide/index.de.html>

[2] https://www4.cs.fau.de/Lehre/WS20/V_BS/Uebungen/seminar-git.pdf

[3] <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>