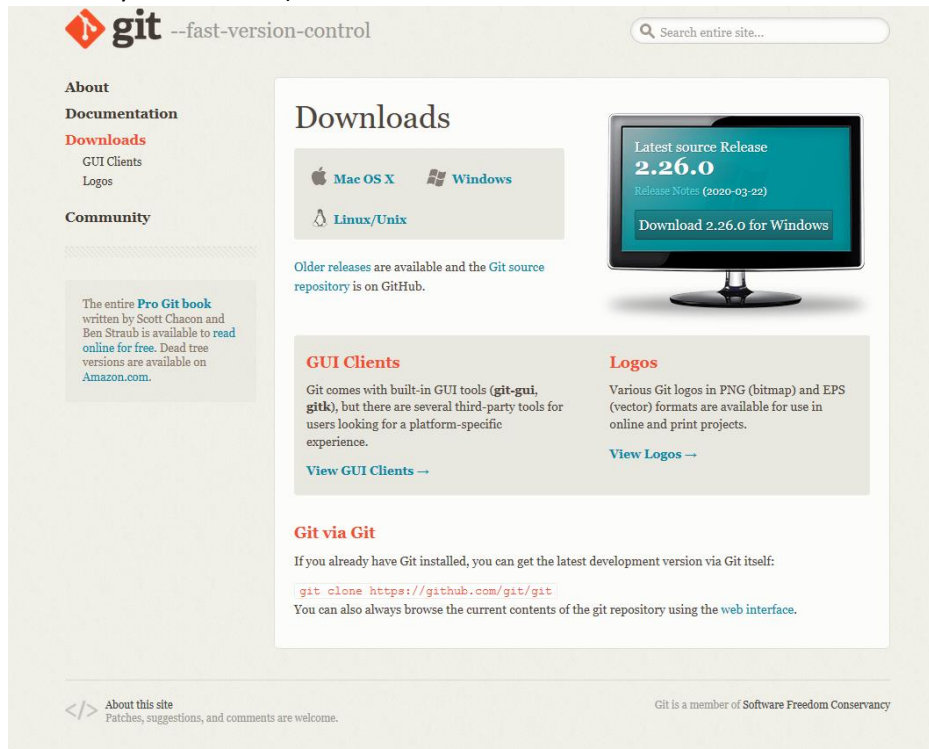


# Installation/Konfiguration/Nutzung von git

## Installation:

1. Git installer von <https://git-scm.com/downloads> herunterladen (den passenden für Ihr Betriebssystem wählen).

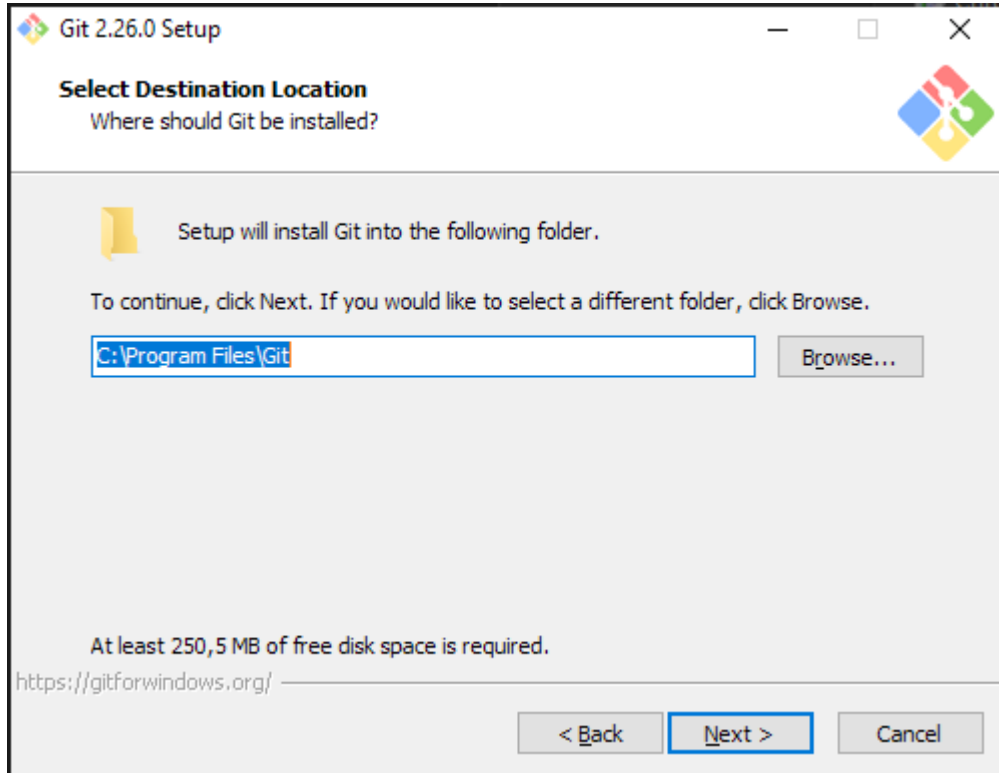


2. Und anschließend ausführen.

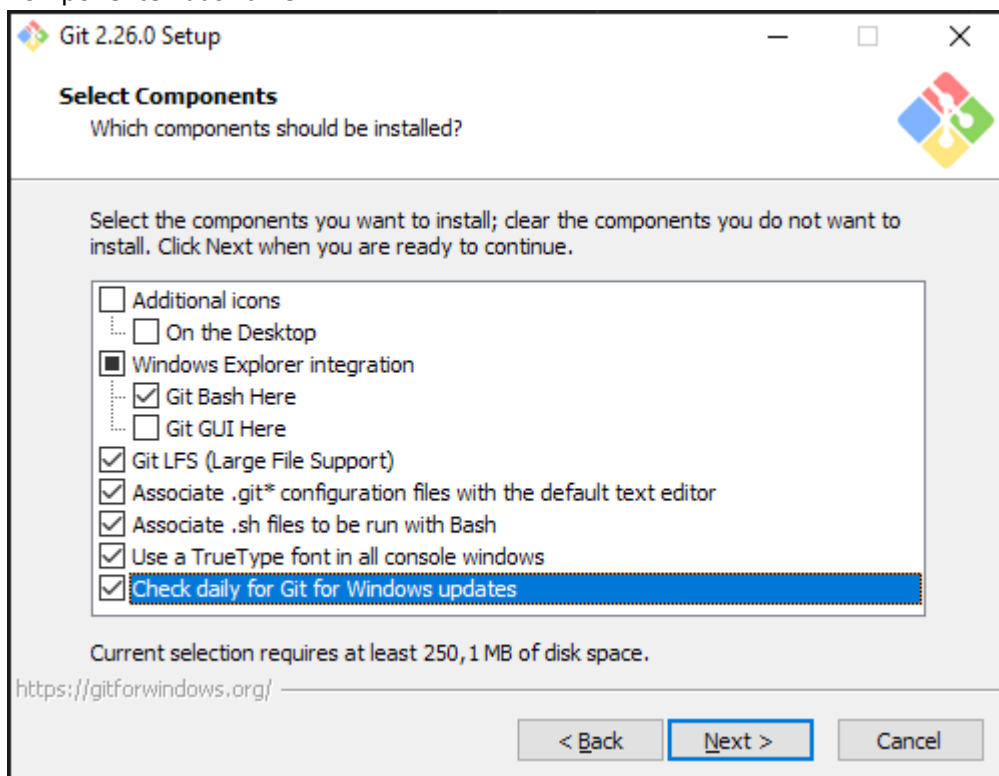
3. Lizenz akzeptieren (Next):



4. Pfad bestätigen (wenn notwendig, ändern):



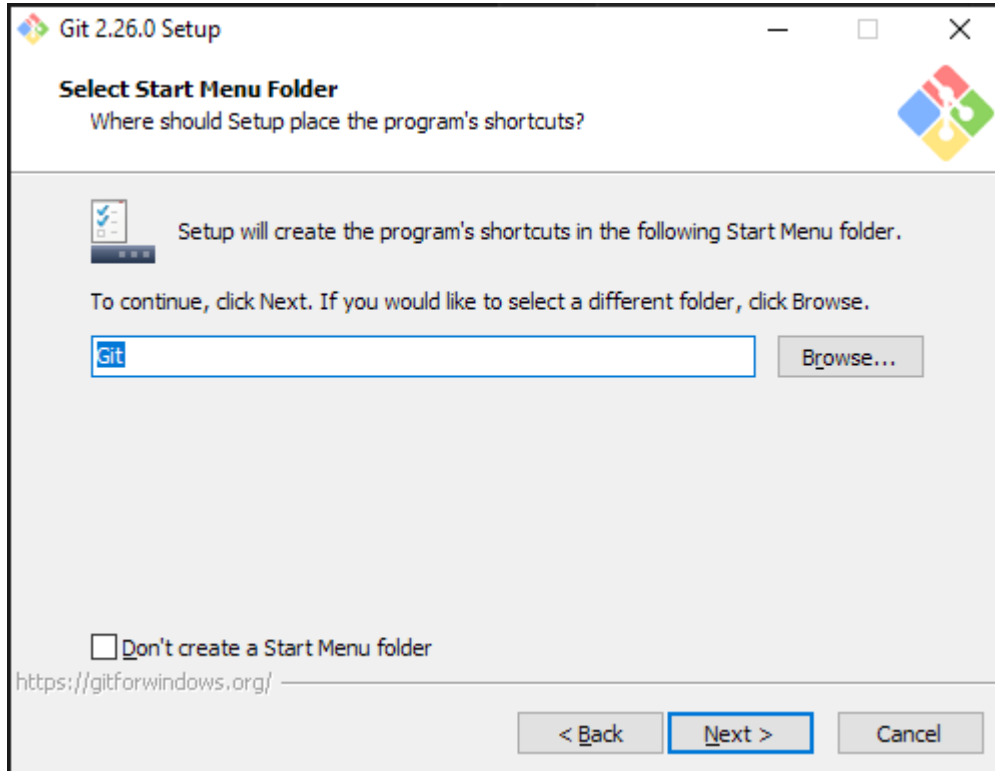
5. Komponenten auswählen:



Ich habe folgende Änderungen vorgenommen:

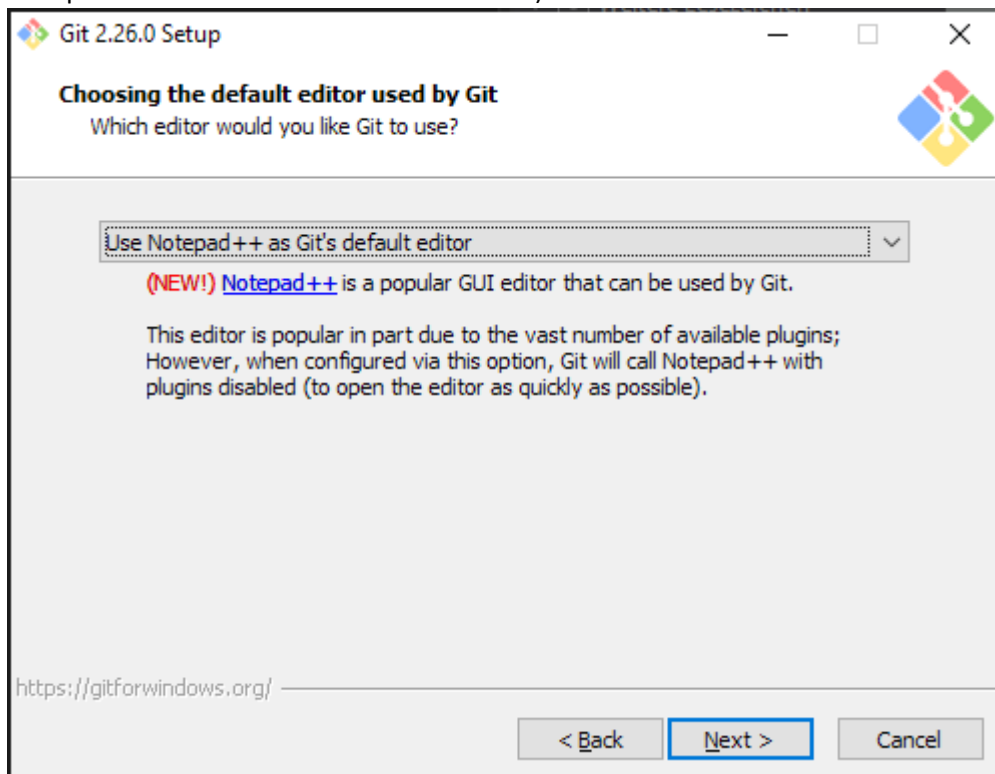
- Git GUI here
- + Use a TrueType font in all console windows
- + Check daily for Git for Windows updates

6. Startmenü Eintrag bestätigen

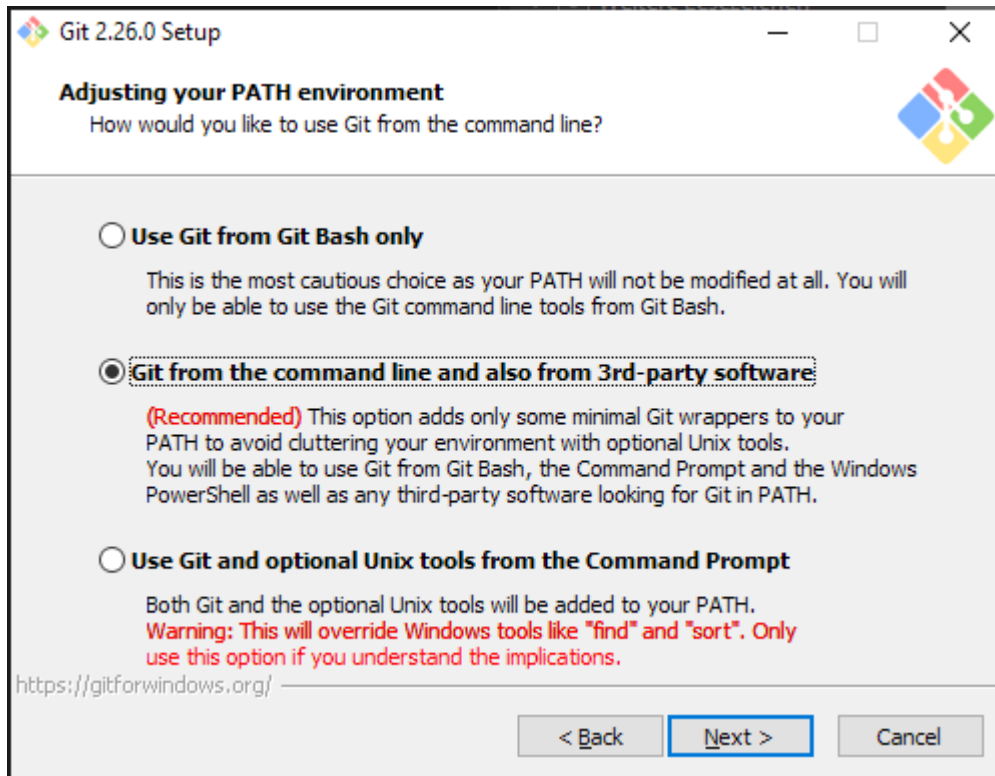


7. Standard Texteditor auswählen:

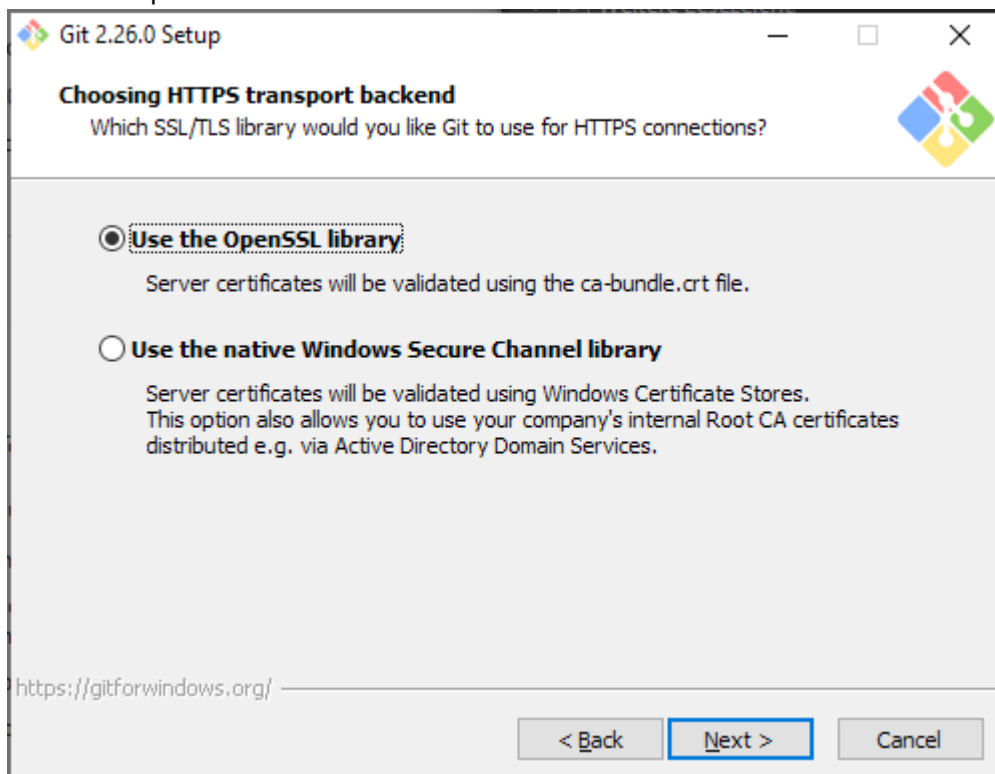
Hier können Sie einen Editor zum Bearbeiten von Commit-Messages auswählen. (Sie können hier einen vorgegebenen auswählen oder einen anderen selbst einstellen. Der gewählte Editor muss aber installiert sein, das wird geprüft. Am einfachsten ist wahrscheinlich Notepad++ zu installieren und auszuwählen.)



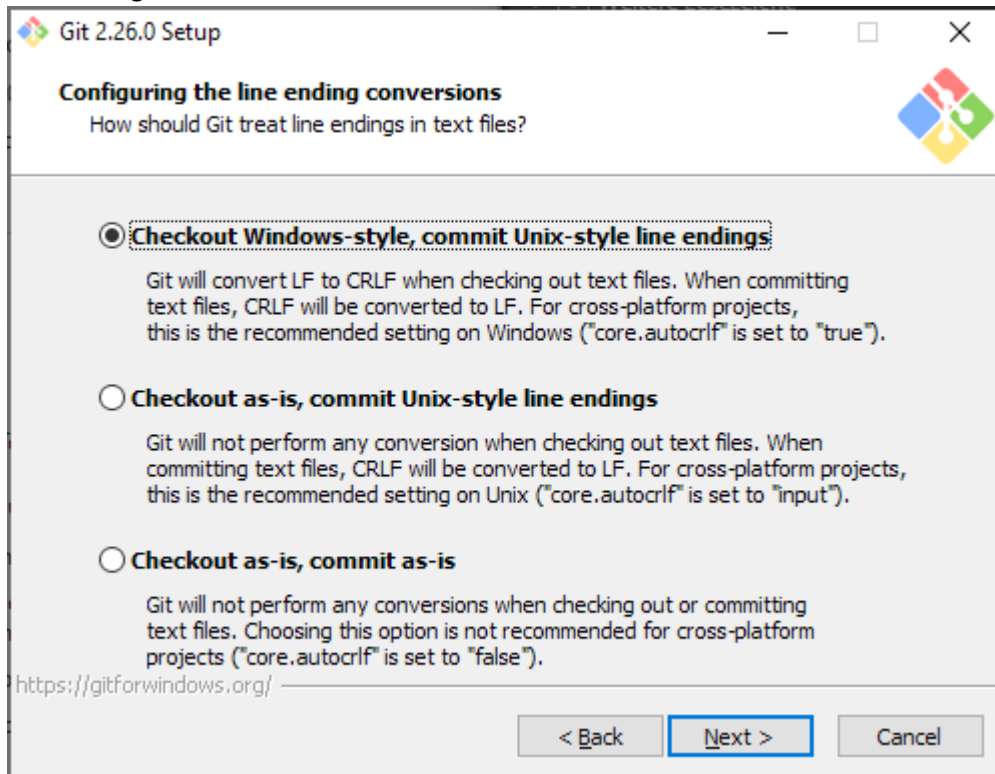
8. Einstellung der Umgebungsvariablen: (Hier ist die vorgegebene Option „Git from the command line and also from 3rd-party software“ die beste Wahl.)



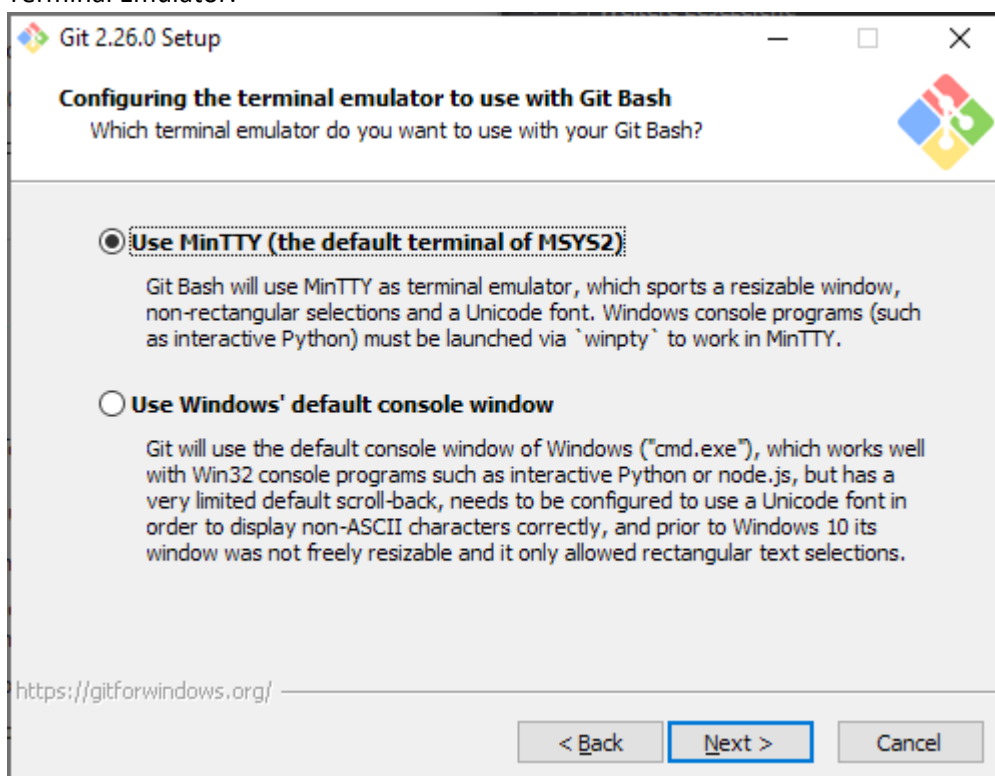
9. HTTPS transport backend:



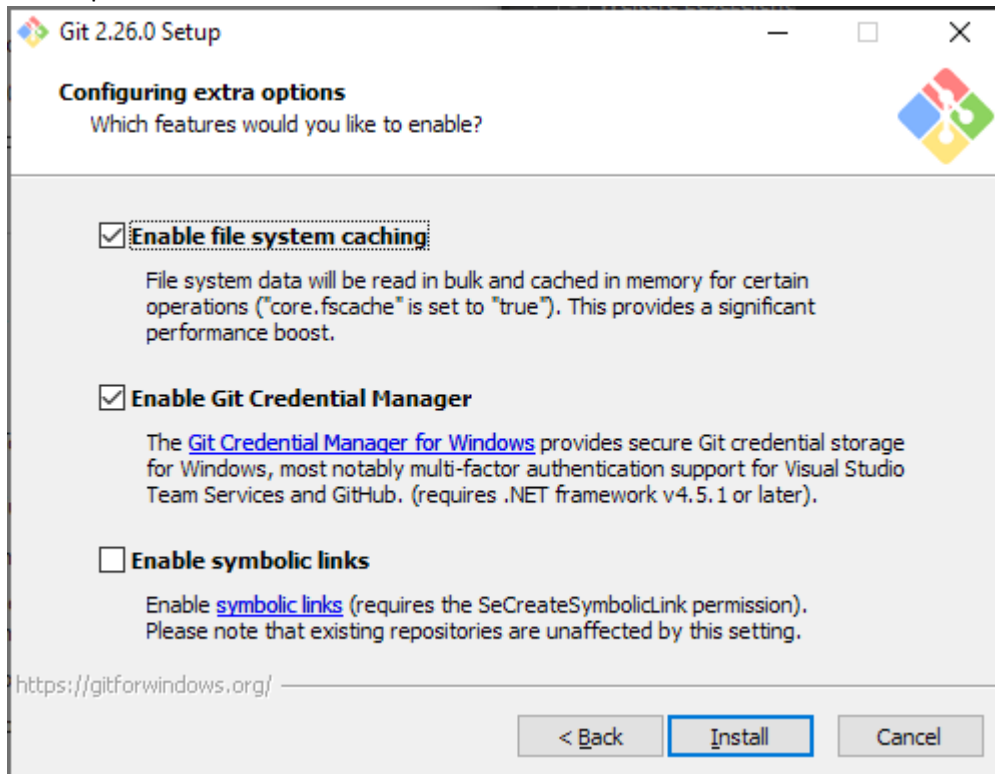
10. Line endings:



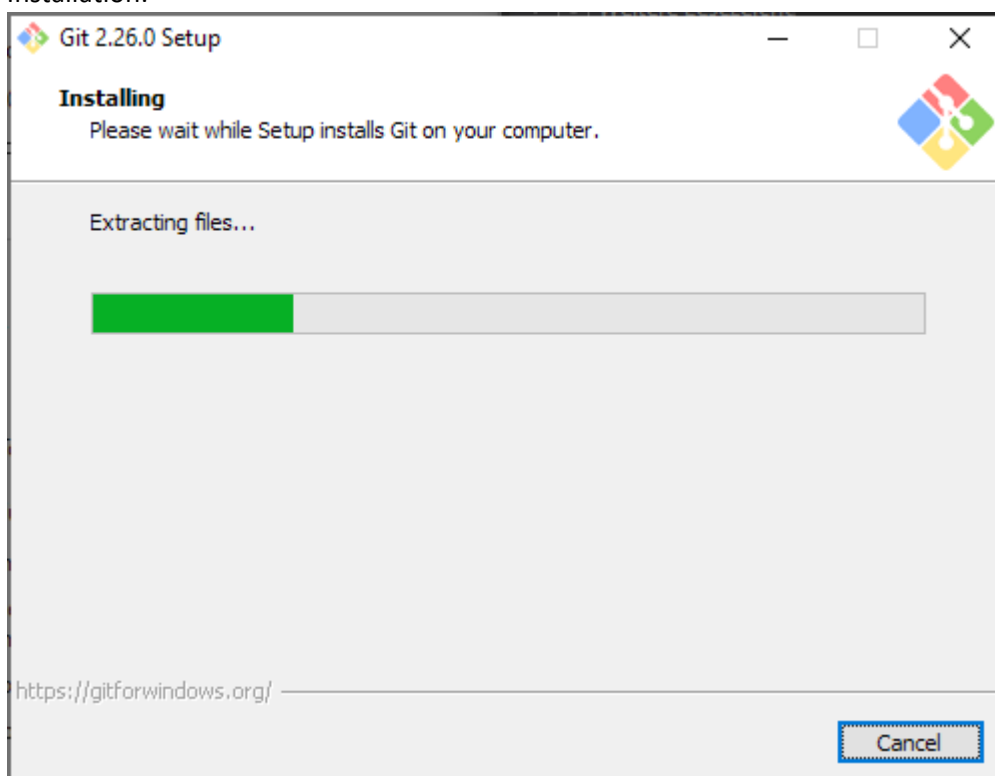
11. Terminal Emulator:



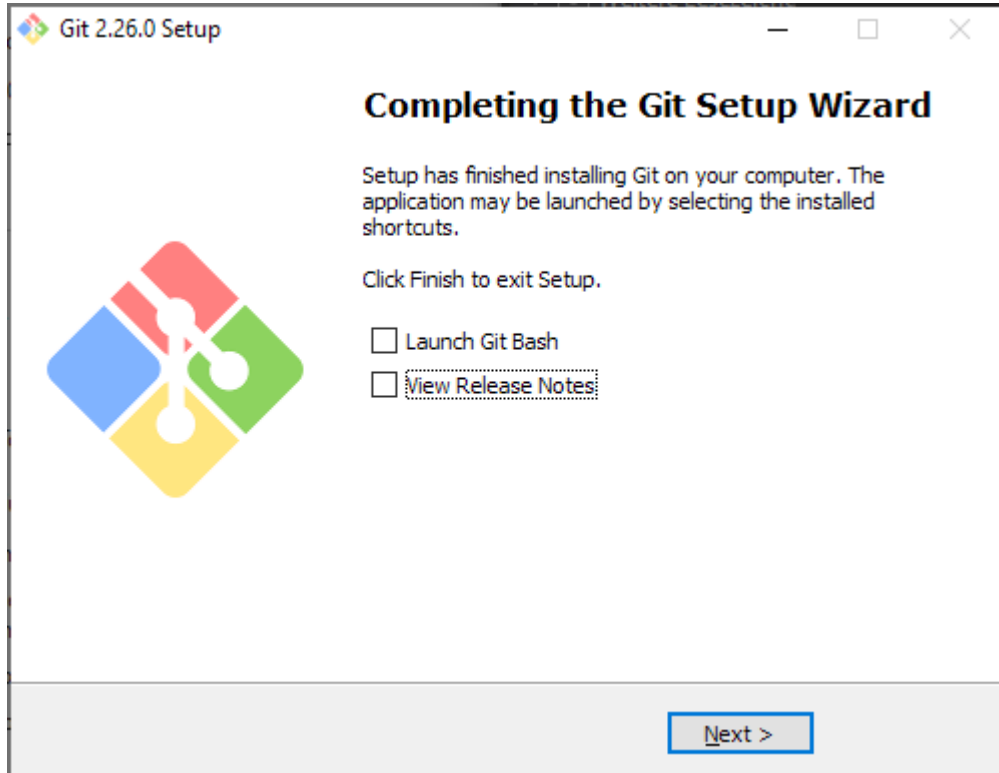
12. Extra options:



13. Installation:

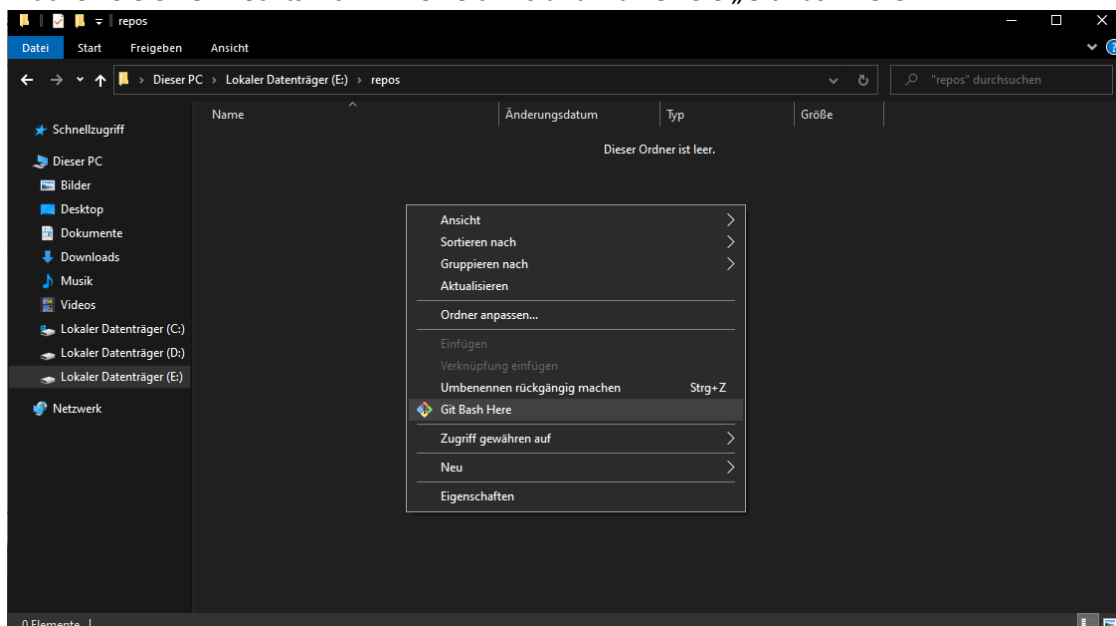


## 14. Abschluss



## Konfiguration

1. Erstellen Sie sich ein Verzeichnis, in dem Sie die Repositories des Lehrgangs ablegen wollen.
2. Wechseln Sie in das Verzeichnis.
3. Machen Sie einen Rechtsklick im Verzeichnis und wählen Sie „Git Bash Here“

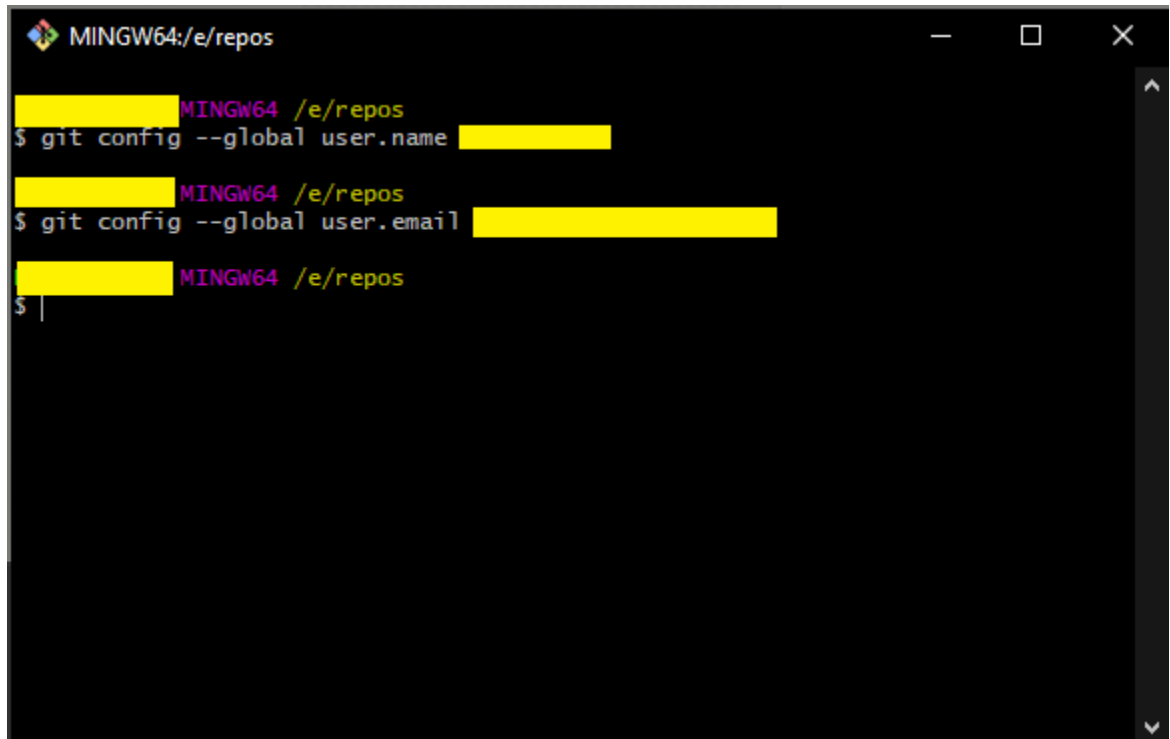


4. Benutzername und eMail festlegen.

Git versieht jeden commit mit einem Benutzernamen und einer eMail-Adresse. Damit diese Zuordnung funktioniert, müssen Sie dies einmal konfigurieren.

Dazu geben Sie folgende Zeilen in die Git Bash ein (Ersetzen Sie <Benutzername> und <eMail> durch Ihre Angaben):

```
git config --global user.name <Benutzername>
git config --global user.email <eMail>
```

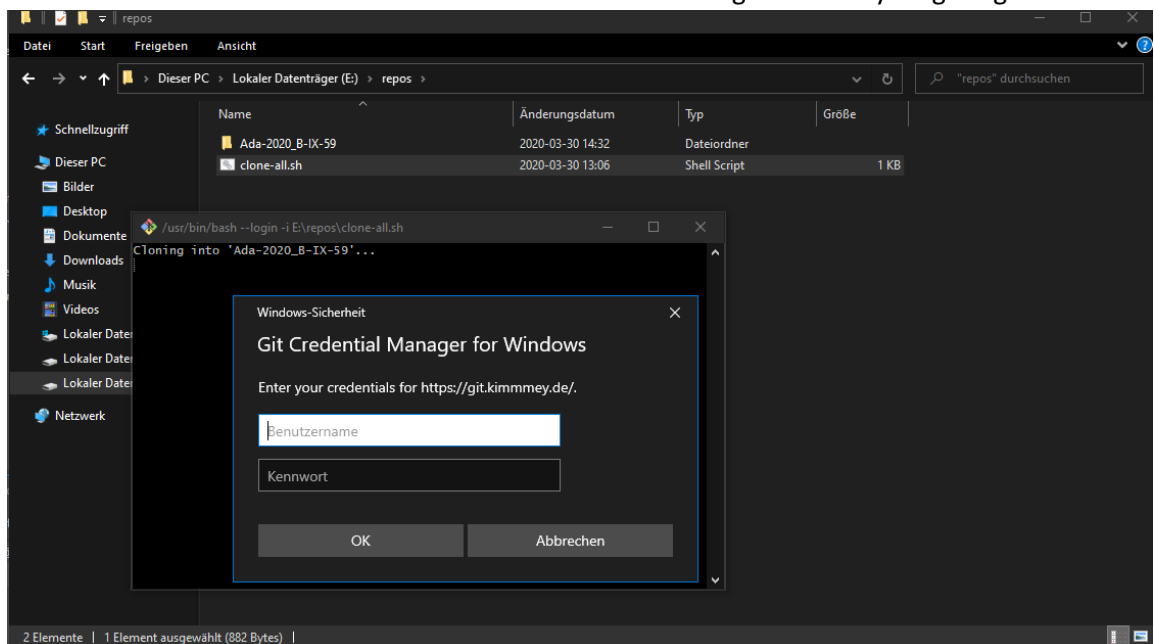


```
MINGW64 /e/repos
$ git config --global user.name [redacted]
$ git config --global user.email [redacted]
$
```

5. Kopieren Sie die Dateien clone-all.sh und pull-all.sh in das Verzeichnis.

Führen Sie anschließend die Datei clone-all.sh aus.

Dabei werden Sie nach einem Nutzernamen und Passwort für git.kimmey.de gefragt:



Geben Sie den Benutzername und das Passwort aus der Whatsapp Nachricht, welche Sie persönlich erhalten haben ein. Nach einem Klick auf OK, werden alle Repositories geladen.



6. Jeder Teilnehmer hat ein eigenes Repository, weil git anders funktioniert als svn. Svn funktioniert Dateibezogen. Git hingegen arbeitet commit bezogen, das heißt Sie können nur etwas hochladen, nachdem sie den aktuellen Stand heruntergeladen haben. Das würde bei 13 Personen mit Zugriff zu Problemen führen, die ich umgehen möchte. Sie haben Lesezugriff auf alle Repositories, allerdings nur auf Ihrem eigenen Schreibzugriff. Das bedeutet: Sie können auf Ihrem PC zwar in jeden der Ordner schreiben, aber auf den Server können Sie nur die Inhalte aus Ihrem Verzeichnis hochladen. Wie das funktioniert sehen Sie im folgenden Abschnitt.

### Repositories aktualisieren

Zum Aktualisieren aller Repositories führen Sie die Datei `pull-all.sh` aus.

Falls Sie nur ein bestimmtes Repository aktualisieren wollen, wechseln Sie in das Verzeichnis. Anschließend machen Sie in einem leeren Bereich und wählen „Git Bash Here“ aus. Nach der Eingabe von „`git pull`“ (ohne Anführungszeichen und gefolgt von Enter) wird das aktuelle Repository aktualisiert.

### Das eigene Repository hochladen

Zum Hochladen der geänderten Dateien, wechseln Sie in Ihr eigenes Verzeichnis ( zum Beispiel: „Ada-2020\_B-IX-59-XX“) und machen einen Rechtsklick im leeren Bereich. Anschließend wählen Sie „Git Bash Here“. Jetzt haben Sie folgende Möglichkeiten:

- `git status` Zeigt Ihnen die veränderten Dateien an. (kann immer ausgeführt werden.)

Hochladen (die Reihenfolge muss eingehalten werden):

- `git add <Datei/Verzeichnis>` Markiert die Datei / das Verzeichnis zum hochladen.
- `git commit -m „<Kommentar>“` Fügt einen Kommentar hinzu. (Notwendig)
- `git push` Veröffentlicht die Änderungen auf dem Server

Tipp:


- `git add .` Markiert alle geänderten Dateien im aktuellen und allen Unterverzeichnissen zum hochladen. (Der Punkt ist wichtig.)

### GUI für git (optional)

Es gibt mehrere grafische Oberflächen, um die Arbeit mit git zu erleichtern. Eine dieser Oberflächen, welche für Windows und Mac verfügbar ist, ist Fork. Diese GUI kann unter <https://git-fork.com/> heruntergeladen werden. Da steht zwar ein Preis von 49,99 \$, aber das Programm kann vollständig und unbegrenzt getestet werden. Da Sie sich das Programm selbst herunterladen und installieren testen Sie es natürlich völlig individuell (und kostenlos).

Nach dem Download muss Fork installiert werden. Dafür folgen Sie den Schritten:

1. ForkInstaller.exe ausführen.
2. User Name und Email ausfüllen. (Dieseangaben beziehen sich wieder auf git commits und den entsprechenden Nutzernamen und die eMail-Adresse. Diese Angaben haben nichts mit evtl. Accounts irgendeiner Webseite zu tun.)
3. Das „Default source folder“ ist das Verzeichnis, in dem Ihre Repositories liegen. Bei mir ist dies (wie oben in den Screenshots zu erkennen) `e:\repos`.




## Welcome to Fork

### User Information

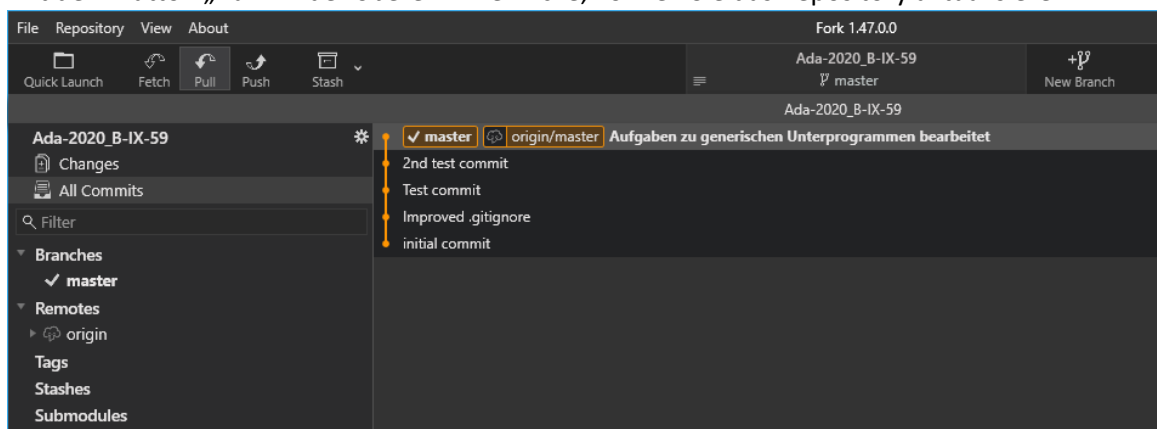
Setup your user name and email address. This information will be associated with your Git commits.

User Name:

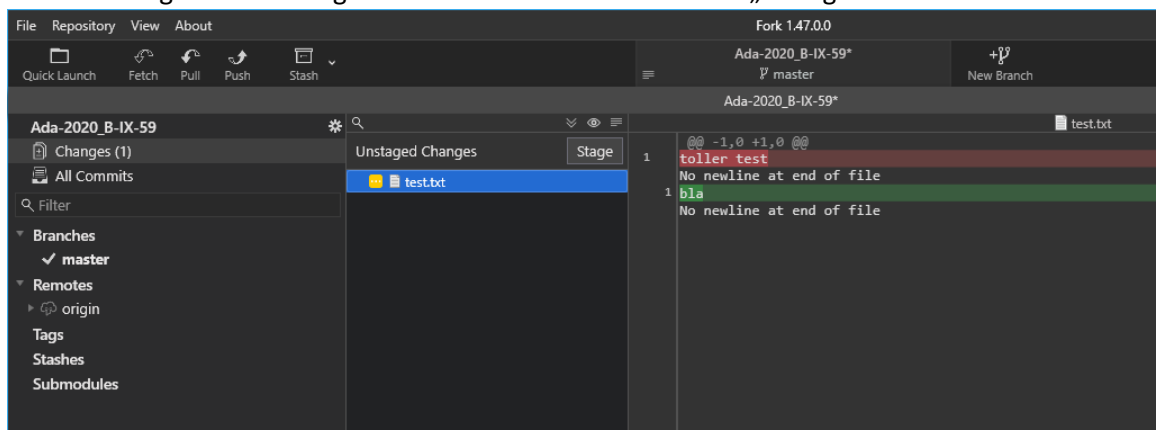
Email:

Default source folder:  

4. Nach einem Klick auf „Finish“ öffnet sich Fork.  
Es werden alle Repositories im angegebenen Verzeichnis automatisch gefunden.
5. Um ein Repository zu öffnen, klicken Sie doppelt auf den Name im „Repository Manager“.  
Anschließend öffnet sich das Repository als Tab.
6. Mit dem Button „Pull“ in der oberen linken Ecke, können Sie das Repository aktualisieren.

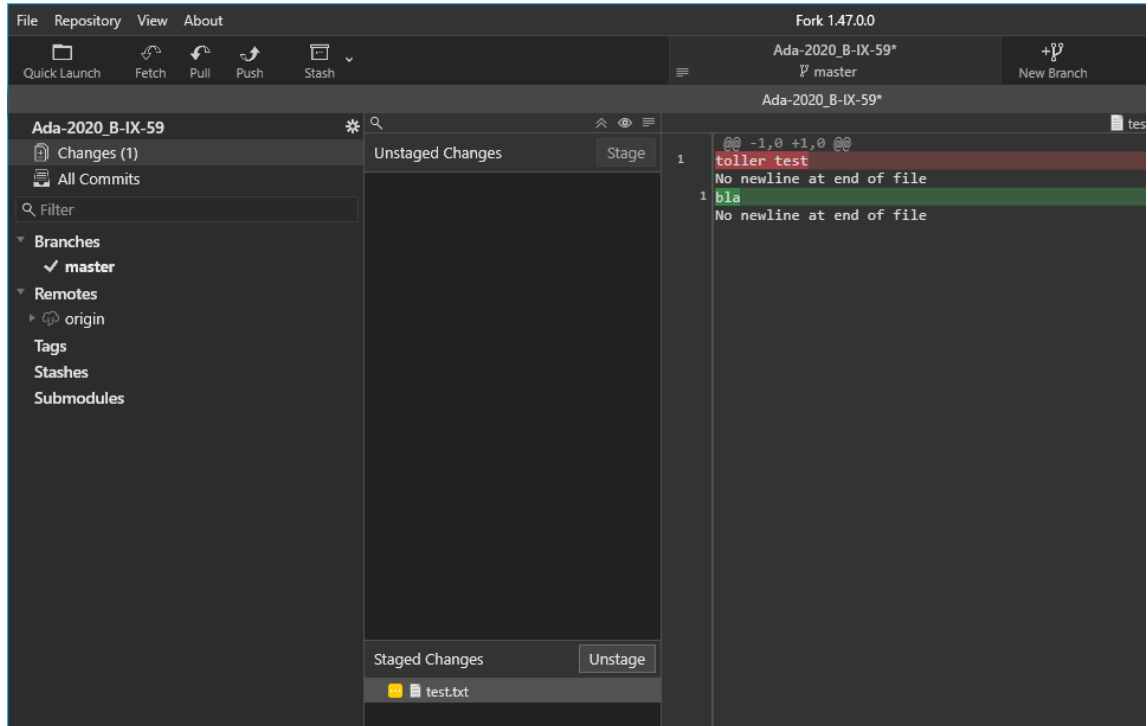


7. Um Änderungen zu übertragen klicken Sie am linken Rand auf „Changes“.

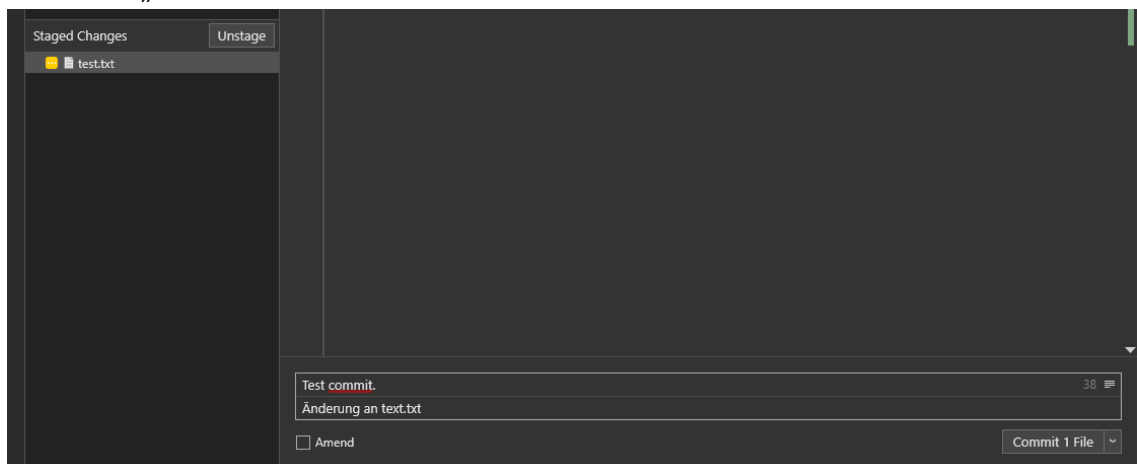


Wählen Sie anschließend alle hochzuladenden Dateien im Bereich „Unstaged Changes“ aus

und klicken Sie „Stage“.



Als nächstes Geben Sie eine Beschreibung in die Textarea unten am Bildschirm ein und klicken auf „Commit“.



Zuletzt klicken Sie in der Menüleiste oben auf „Push“, dadurch wird der Commit auf den Server geladen und steht für andere zum Download (pull) bereit.

