

Tutorial zum Bearbeiten der Acagamics Dokumente

Dieses Dokument ist nur für Organisatoren und Authorisierte bestimmt.

Was beinhaltet diese Datei?

Inhaltsverzeichnis

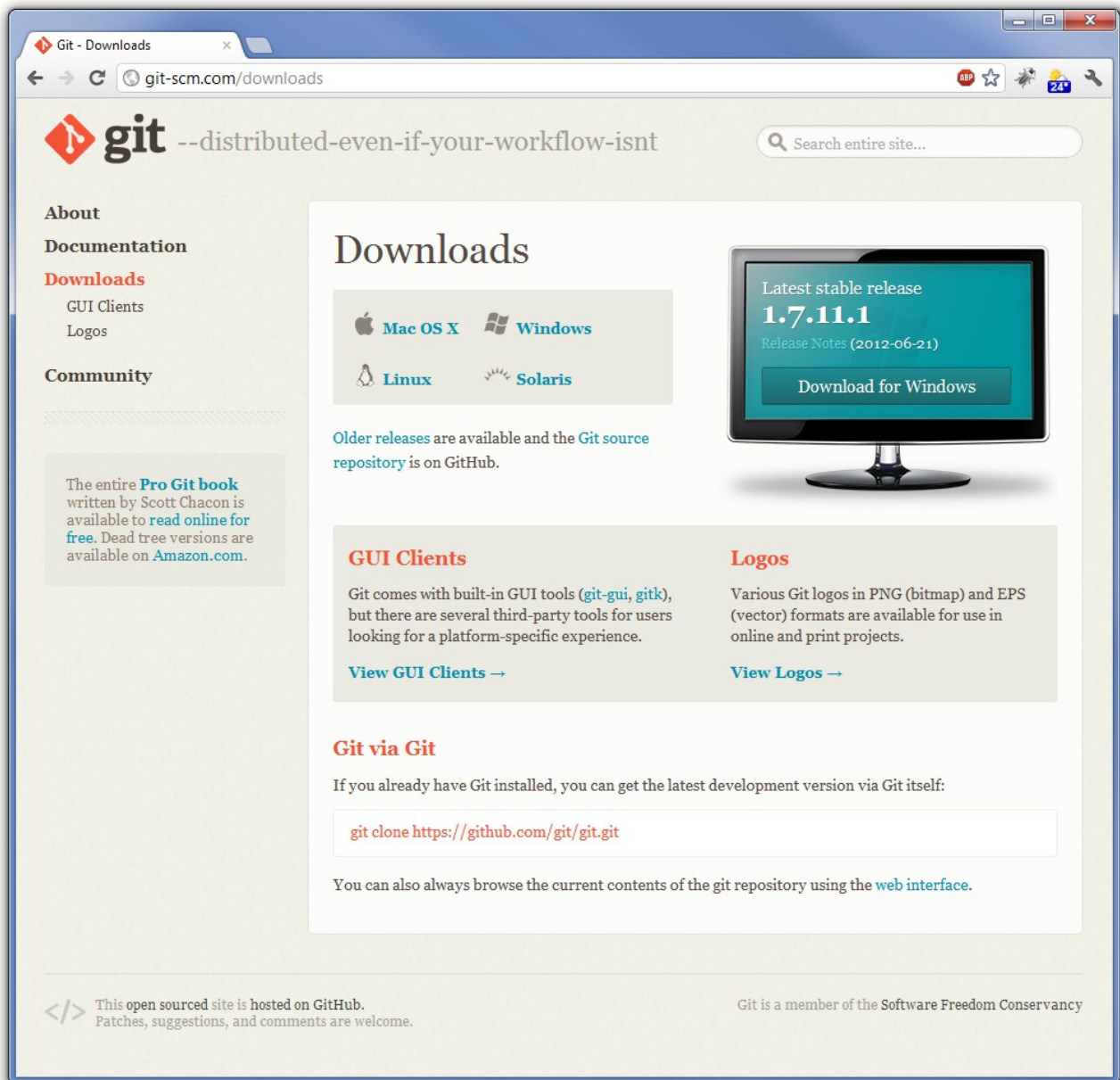
1. Git installieren.....	2
Git download.....	2
2. Git konfigurieren und Key erstellen.....	7
Key erstellen (Ihr habt schon einen Key? Schaut unter Key einsetzen).....	9
Key einsetzen.....	11
Den Key an den Admin senden.....	14
3. Tortoise Git installieren und .ppk Key erstellen.....	15
.ppk Datei erstellen.....	18
4. Tortoise Git verwenden.....	22
Repo mit eurem PC syncen.....	22
Dateien bearbeiten und commiten.....	26

1. Git installieren

Git download

Hierzu geht man auf die Homepage von Git.

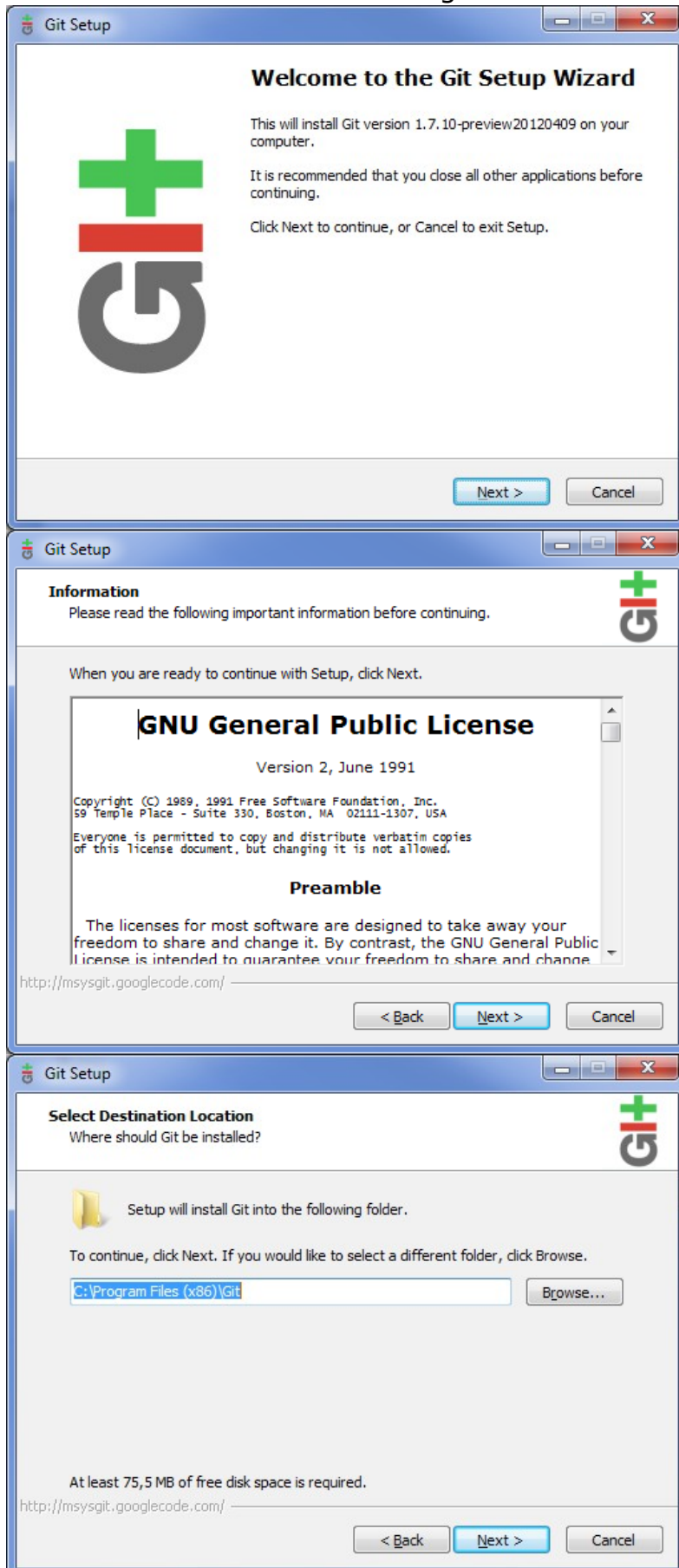
<http://git-scm.com/downloads>

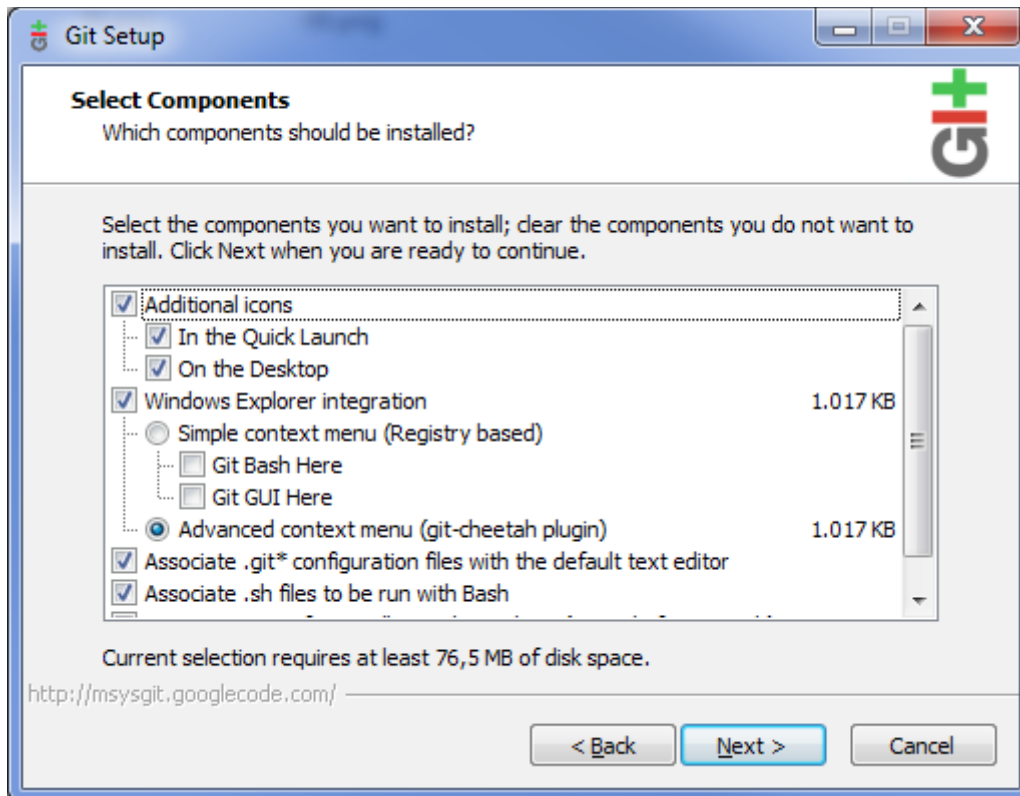


Dort das System anklicken das verwendet wird. Die datei fertig herunterladen und ausführen.

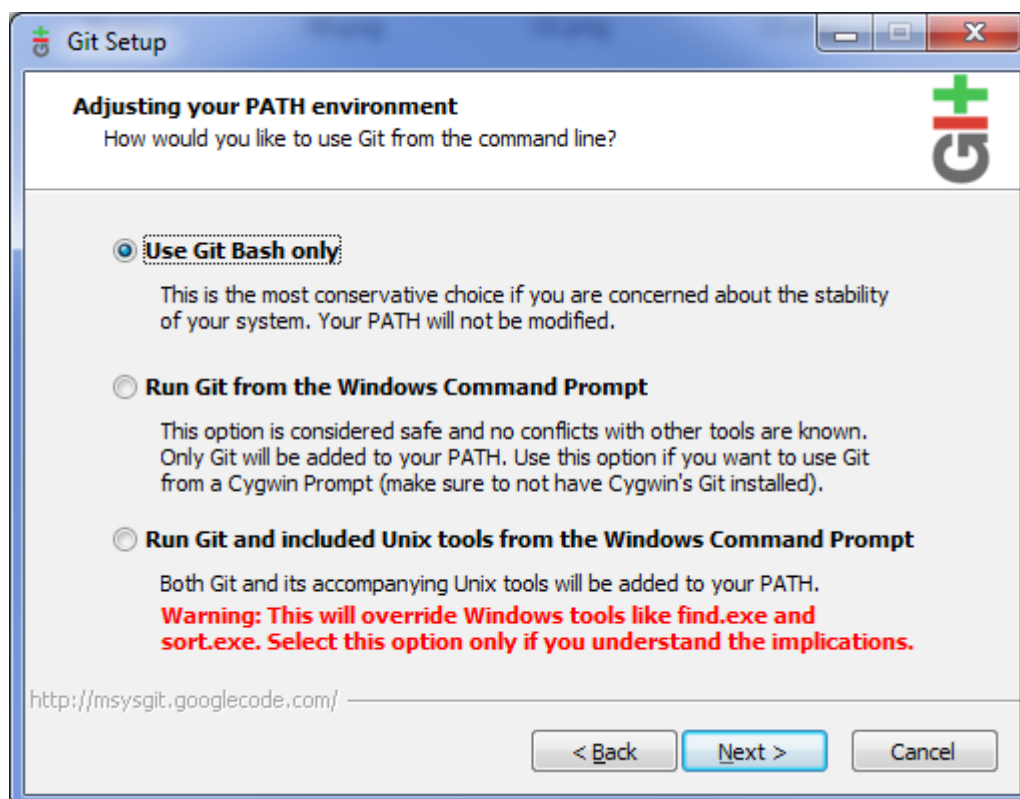
Git installieren

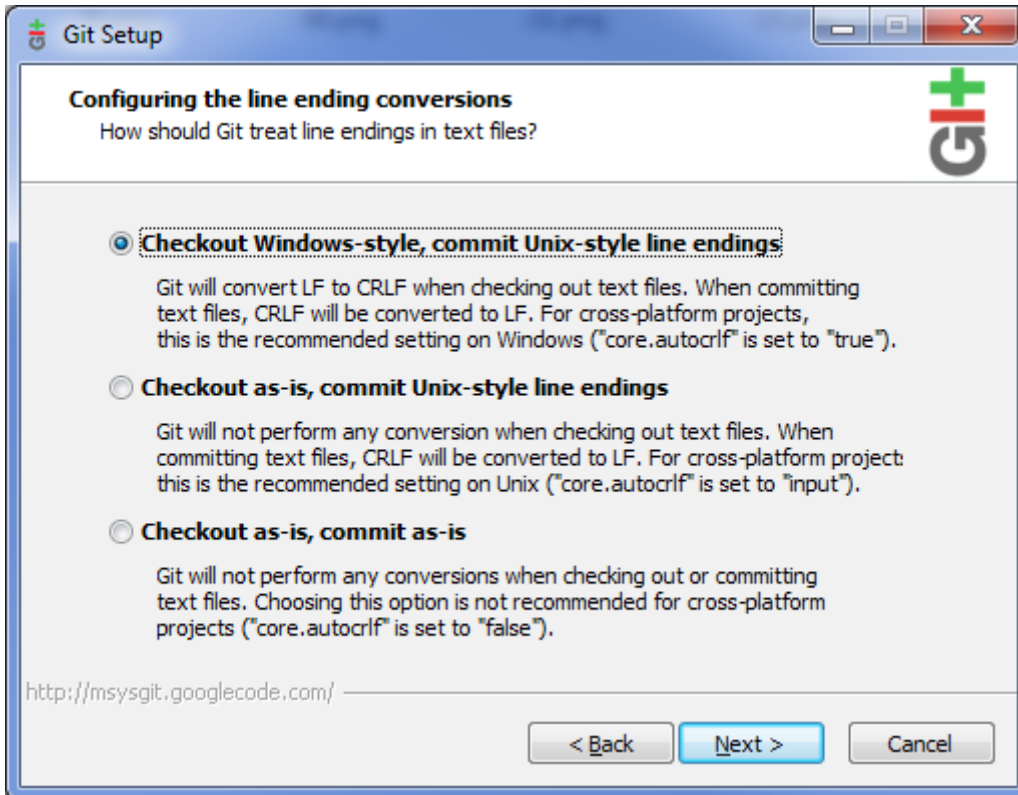
Nun einfach der Bildstrecke folgen...



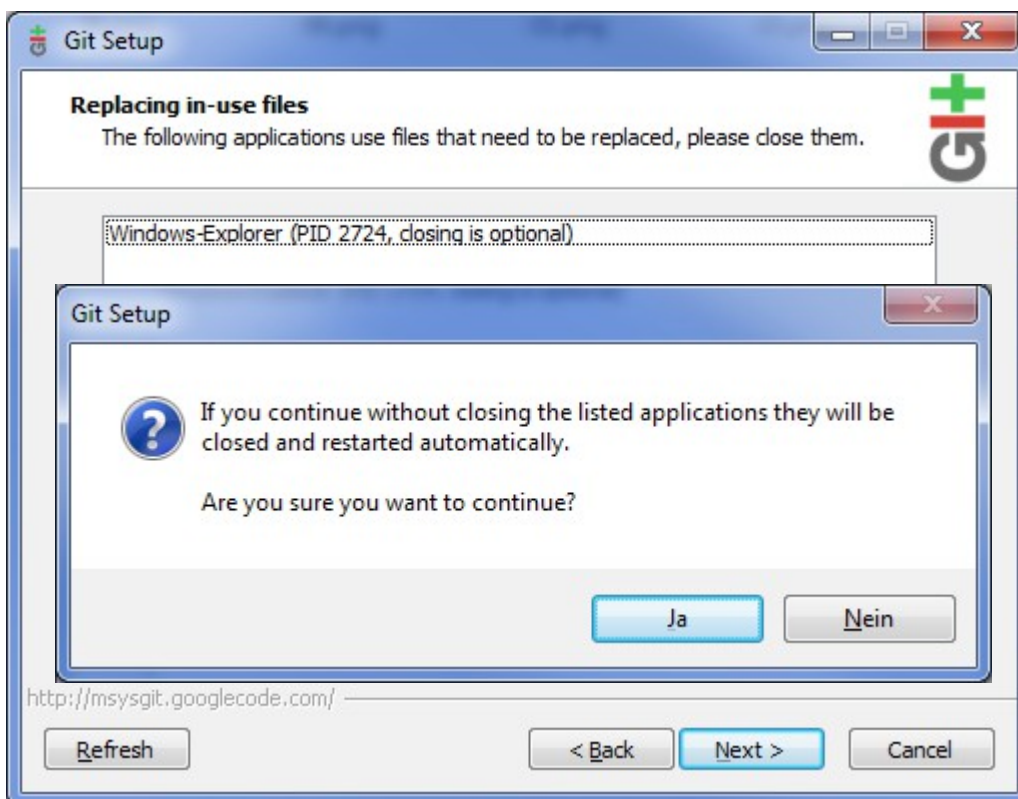


Insert Picture für Startmenu

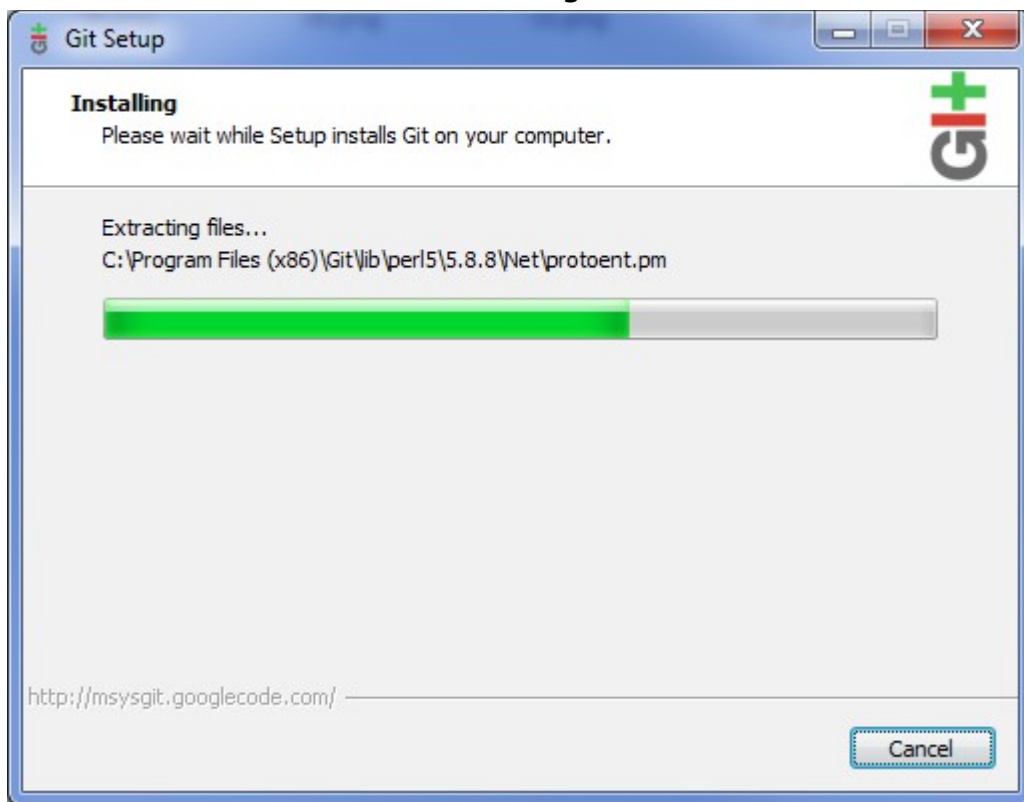




Möglicherweise darf man dann noch Programme schließen.



Und dann sind wir auch schon fertig...

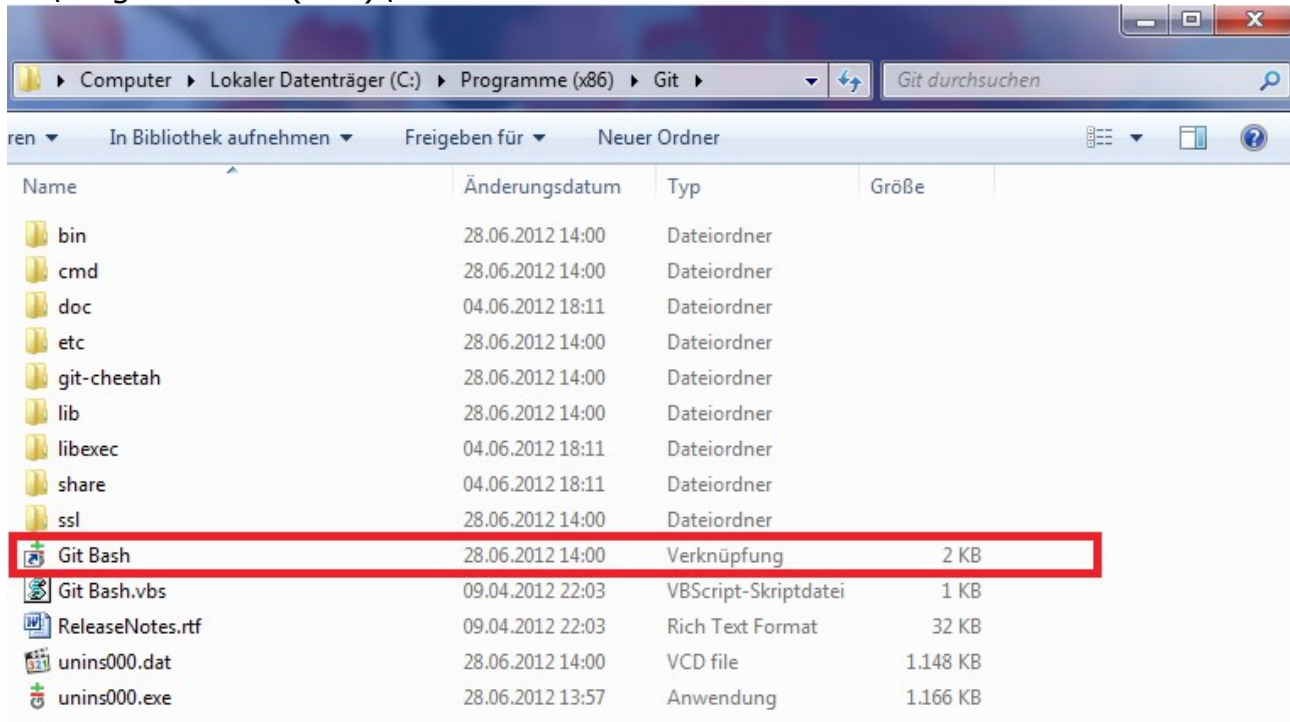


2. Git konfigurieren und Key erstellen

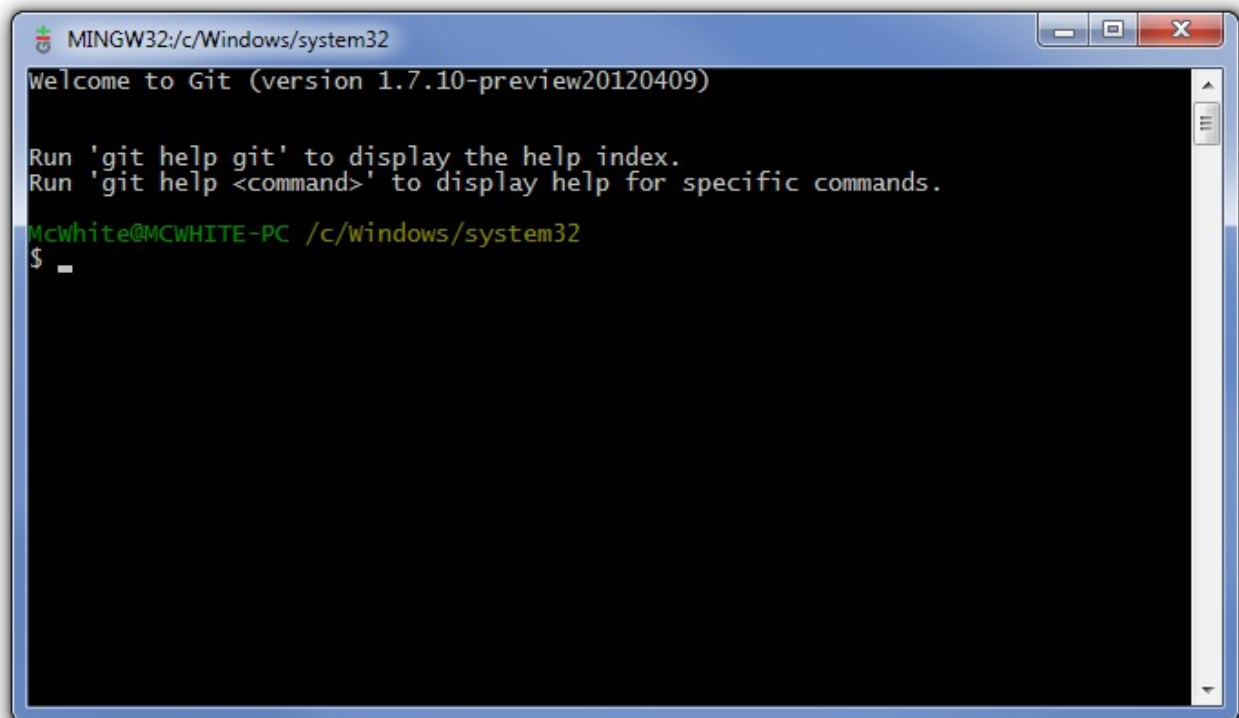
Wenn ihr GIT installiert habt, habt ihr nun auch die Git-Bash.

Diese findet ihr unter Windows hier:

C:\Program Files (x86)\Git



Wenn ihr diese öffnet, seht ihr folgendes:

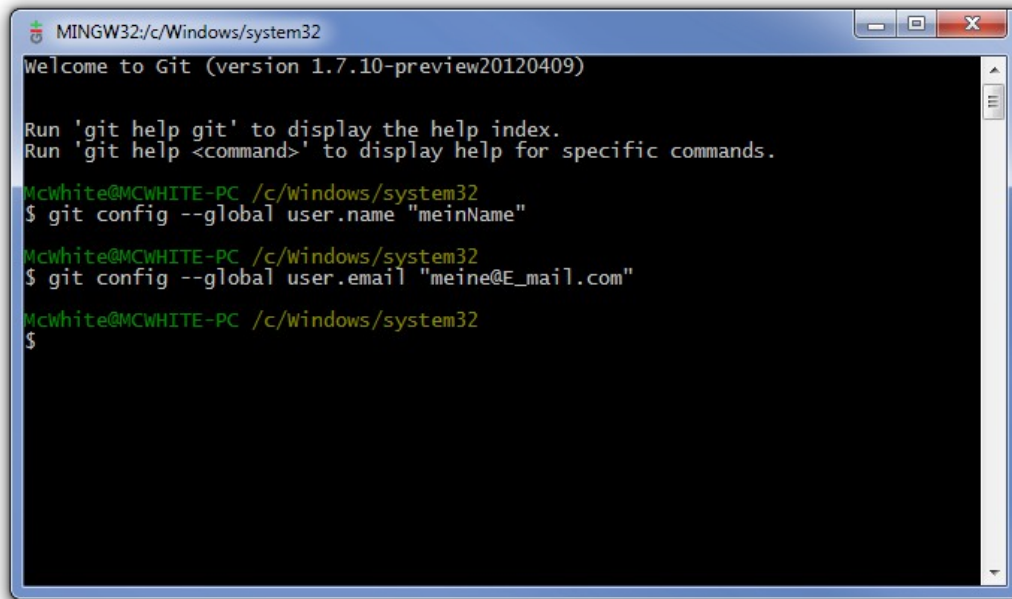


Als erstes erzählt ihr GIT wer ihr seid indem ihr folgendes eingibt und danach enter drückt:

```
git config --global user.name "Hier Euren Namen Eintragen"
```

Danach configuriert ihr euer E-Mail Adresse, diese muss aber nicht unbedingt einer realen Adresse gleichen...

```
git config --global user.email "Name@E_Mail.blubb"
```

A screenshot of a MINGW32 terminal window. The title bar shows the path 'MINGW32/c/Windows/system32'. The terminal output is as follows:

```
Welcome to Git (version 1.7.10-preview20120409)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

McWhite@MCWHITE-PC /c/Windows/system32
$ git config --global user.name "meinName"

McWhite@MCWHITE-PC /c/Windows/system32
$ git config --global user.email "meine@E_mail.com"

McWhite@MCWHITE-PC /c/Windows/system32
$
```

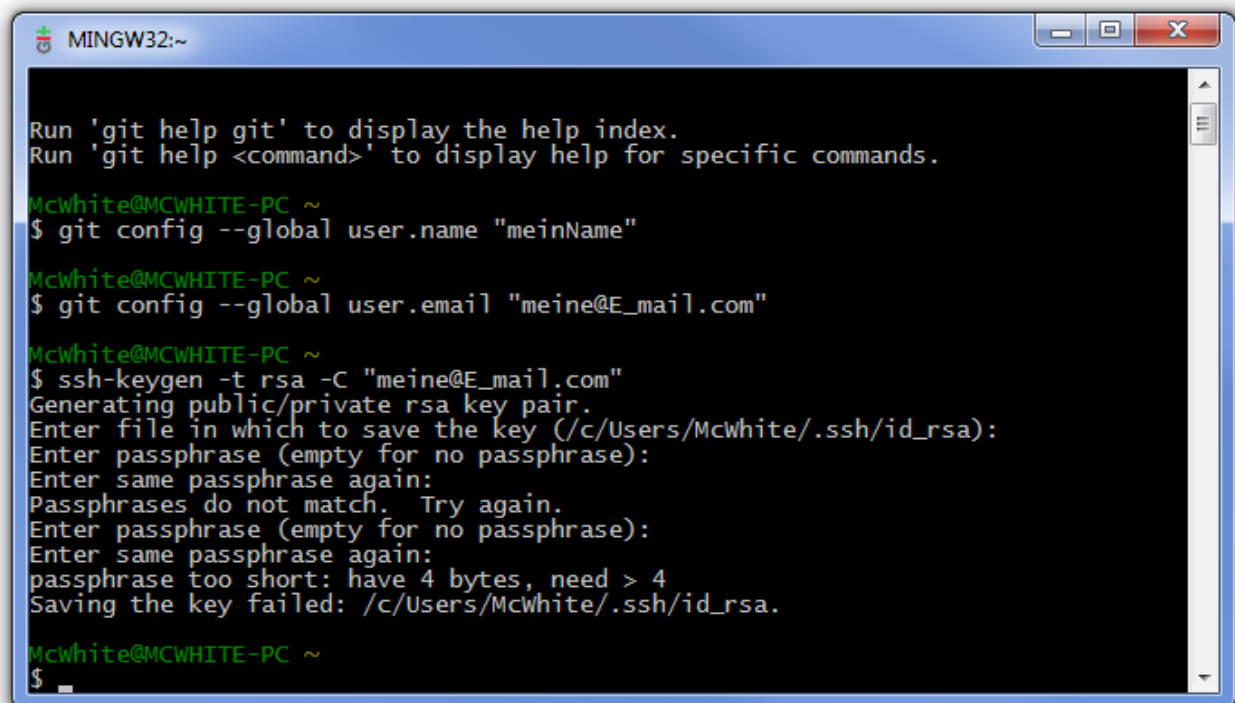

Key erstellen (Ihr habt schon einen Key? Schaut unter **Key einsetzen**)

Nun werden wir einen Key erstellen. Als erstes geben wir folgendes ein:

```
ssh-keygen -t rsa -C "eure_email@eureEmail.blubb"
```

Wenn er daraufhin fragt "Enter file in which to save key..." einfach Enter drücken.

Dann wird er euch sagen "Enter Passphrase", das ist nun euer eigentliches Passwort. Gebt es ein. Dann Gebt es noch einmal ein. Falls ihr euch vertippt fragt er solange bis ihr es hinbekommen habt, außerdem muss das Passwort größer sein als 4 Zeichen, sonst funktioniert es nicht. Hier einmal ein Beispiel bei dem ich alles falsch gemacht habe.



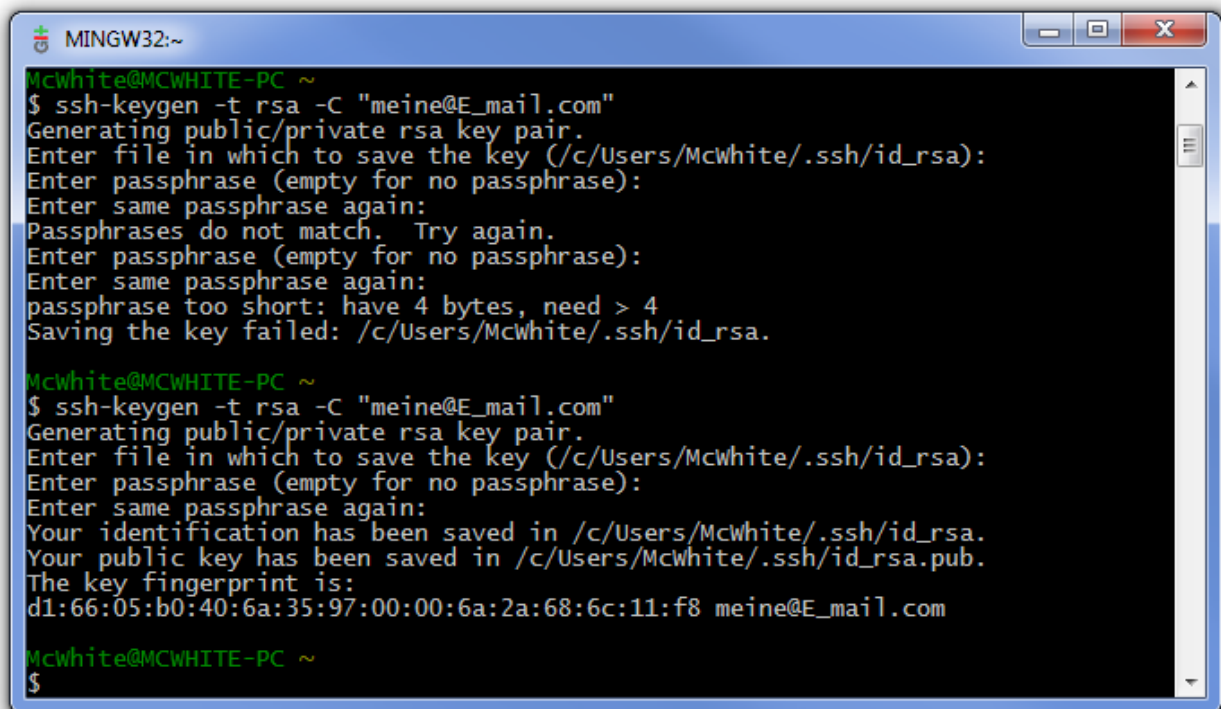
```
MINGW32:~
Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

McWhite@MCWHITE-PC ~
$ git config --global user.name "meinName"

McWhite@MCWHITE-PC ~
$ git config --global user.email "meine@E_mail.com"

McWhite@MCWHITE-PC ~
$ ssh-keygen -t rsa -C "meine@E_mail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/McWhite/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
passphrase too short: have 4 bytes, need > 4
Saving the key failed: /c/Users/McWhite/.ssh/id_rsa.

McWhite@MCWHITE-PC ~
$
```



```
MINGW32:~
McWhite@MCWHITE-PC ~
$ ssh-keygen -t rsa -C "meine@E_mail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/McWhite/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
passphrase too short: have 4 bytes, need > 4
Saving the key failed: /c/Users/McWhite/.ssh/id_rsa.

McWhite@MCWHITE-PC ~
$ ssh-keygen -t rsa -C "meine@E_mail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/McWhite/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/McWhite/.ssh/id_rsa.
Your public key has been saved in /c/Users/McWhite/.ssh/id_rsa.pub.
The key fingerprint is:
d1:66:05:b0:40:6a:35:97:00:00:6a:2a:68:6c:11:f8 meine@E_mail.com

McWhite@MCWHITE-PC ~
$
```

Und so sollte es aussehen wenn ihr alles richtig gemacht

Und den dazugehörigen .ppk Schlüssel. Solltet ihr keinen solchen haben, aber mit Git arbeiten wollen, schaut bitte unter ".ppk Datei erstellen".

Den rsa schlüssel und den .ppk Schlüssel solltet ihr in den Ordner

C:\Users\EuerBenutzername\.ssh\

oder

C:\Benutzer\EuerBenutzername\.ssh\

kopieren. Falls der Ordner noch nicht existiert solltet ihr ihn anlegen.

Ihr könnt nun zum Punkt "Turtle Git" verwenden springen. Wenn ihr nach einem .ppk Key gefragt werdet, nehmt den, den ihr gerade in den .ssh Ordner verfrachtet habt. ;)

Den Key an den Admin senden

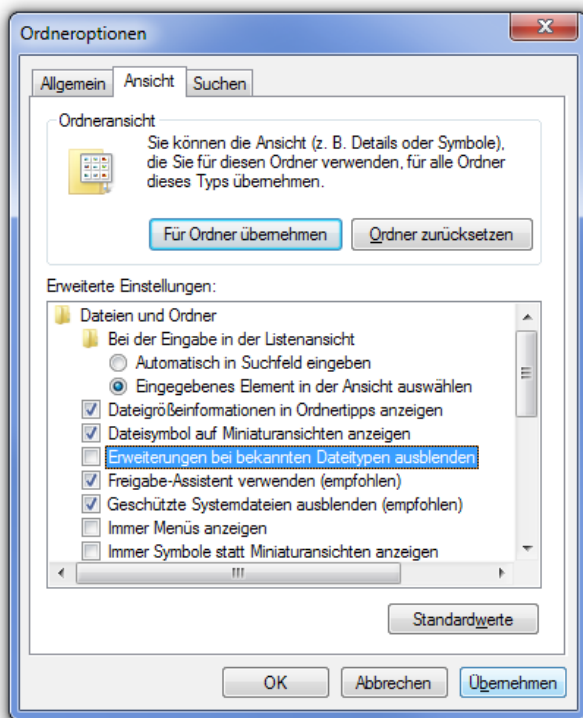
Der Key wurde unter:

C:\Users\Euer_User_Name\.ssh
gespeichert.



Ihr nehmt nun die Datei "id_rsa.pub" und benennt sie in "NameNachname.pub" um.

<-----Optional----->



Solltet ihr nur 2 "ununterscheidbare" Dateien in dem Ordner sehen die "id_rsa" heißen, dann seht ihr die Endungen von bekannten Dateitypen unter Windows nicht, dies könnt ihr ändern indem ihr im Windows Explorer kurz die Taste "alt" antippt, auf Extras geht, dann in die Ordneroptionen unter Ansicht. Dort das Häkchen bei "Erweiterungen bei bekannten Dateitypen ausblenden" rausnehmen und auf Übernehmen klicken.

</-----Optional----->

Als nächstes sendet ihr eine E-Mail mit eurer NameNachname.pub im Anhang an [Insert_E-Mail].

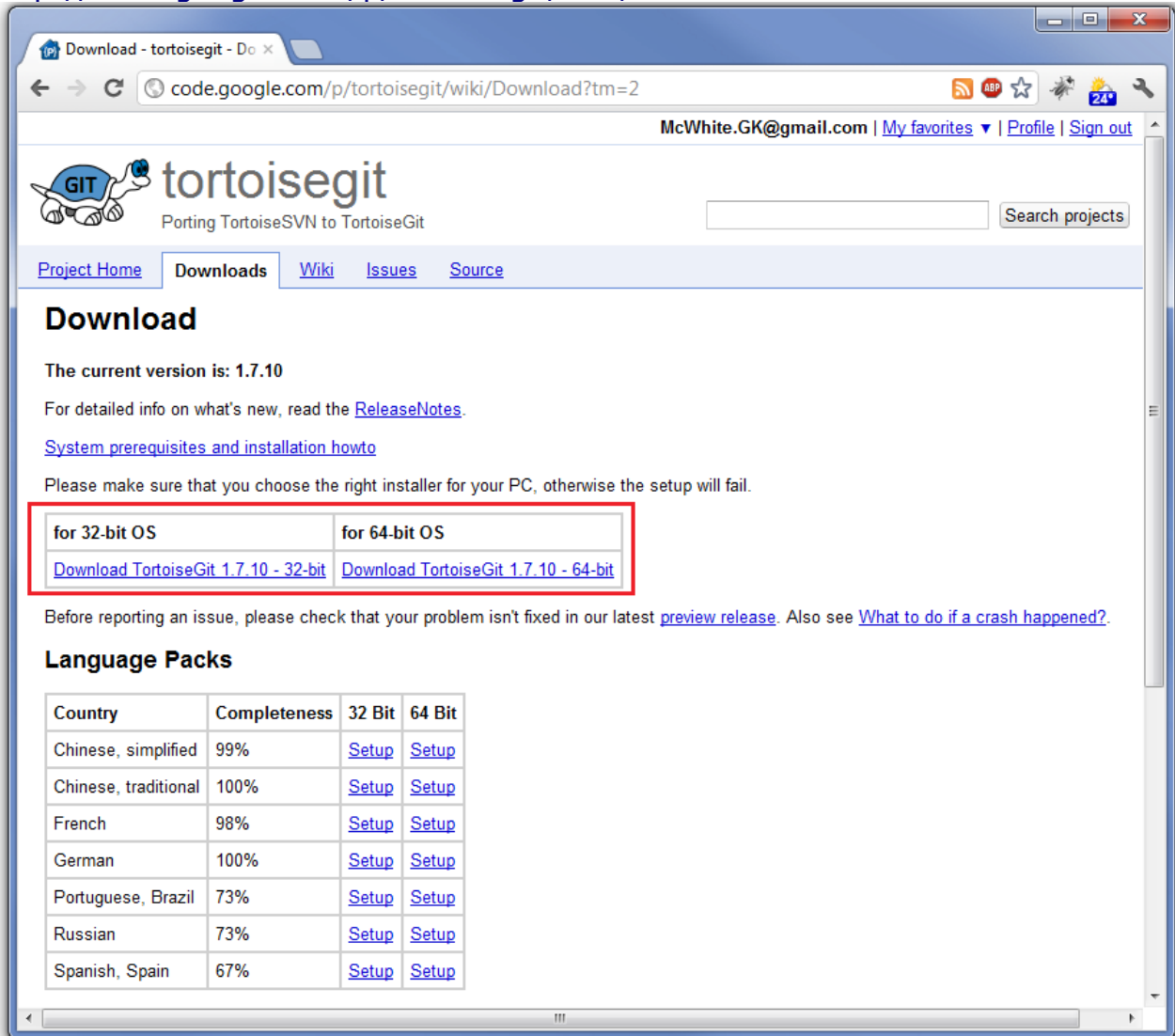
3. Tortoise Git installieren und .ppk Key erstellen

Während ihr auf die Antwort des Admin für eueren Key wartet, können wir uns nun Tortoise Git widmen.

Dieses verwenden wir um den Umgang mit Git zu erleichtern.

Den Installer kann man sich hier besorgen:

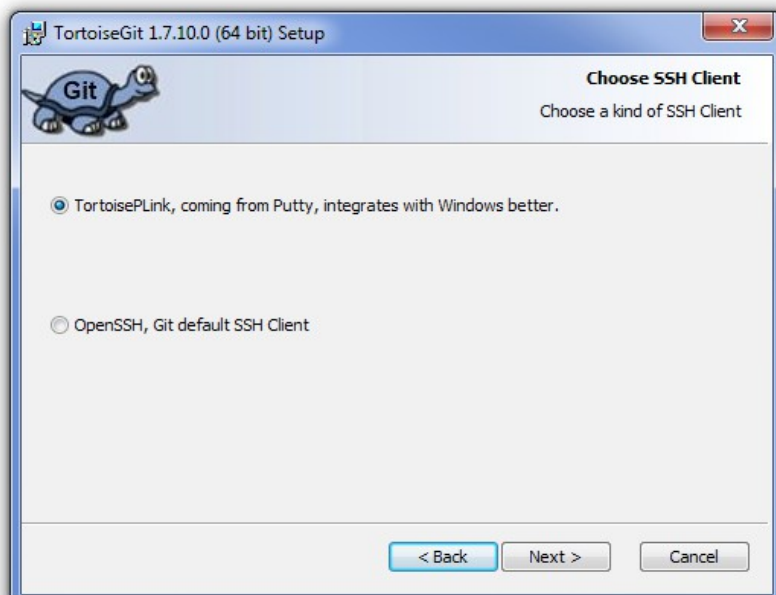
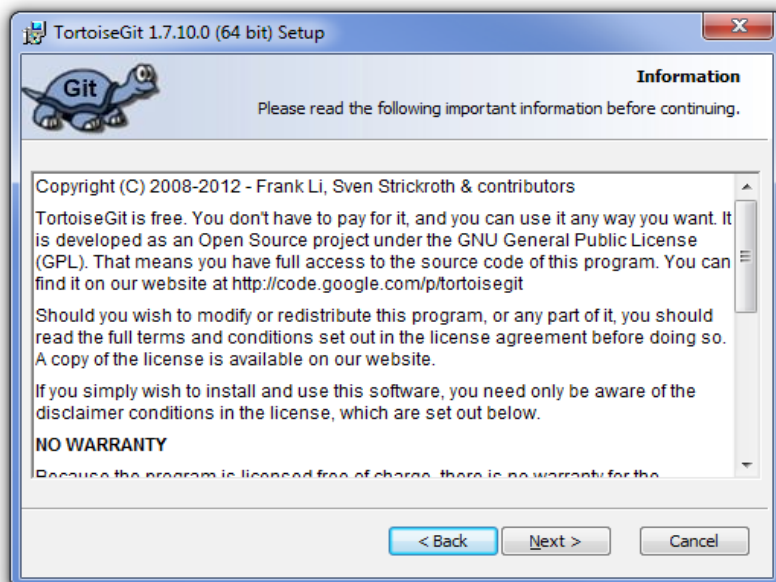
<http://code.google.com/p/tortoisegit/wiki/Download?tm=2>

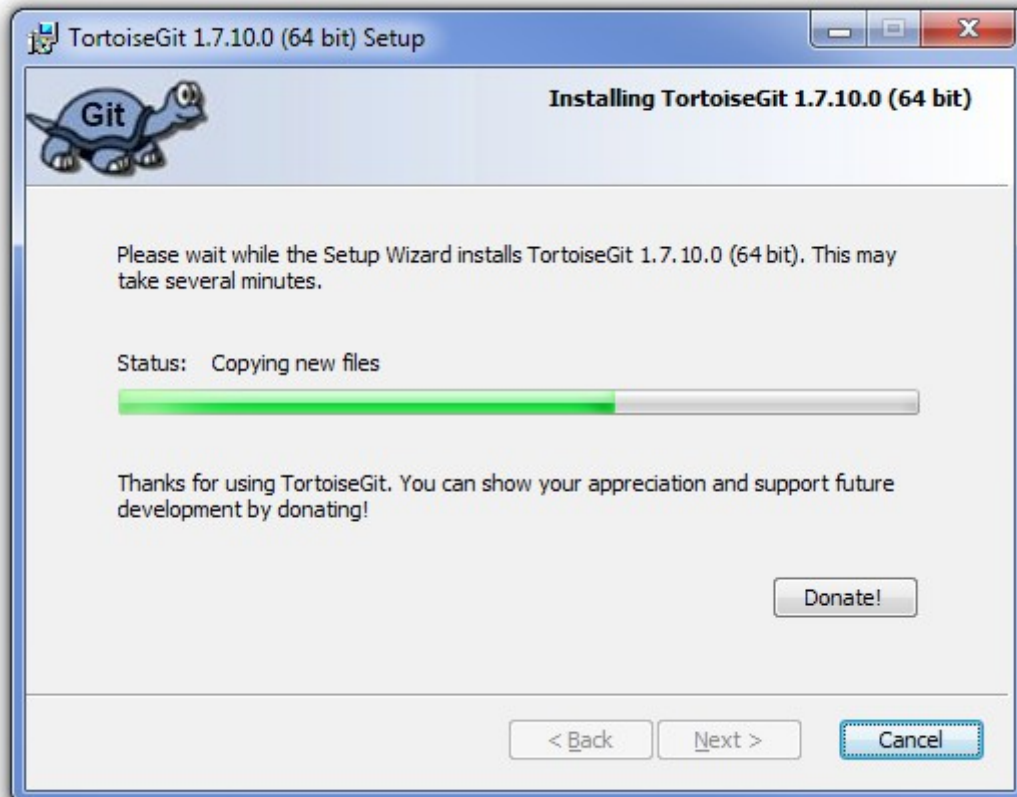
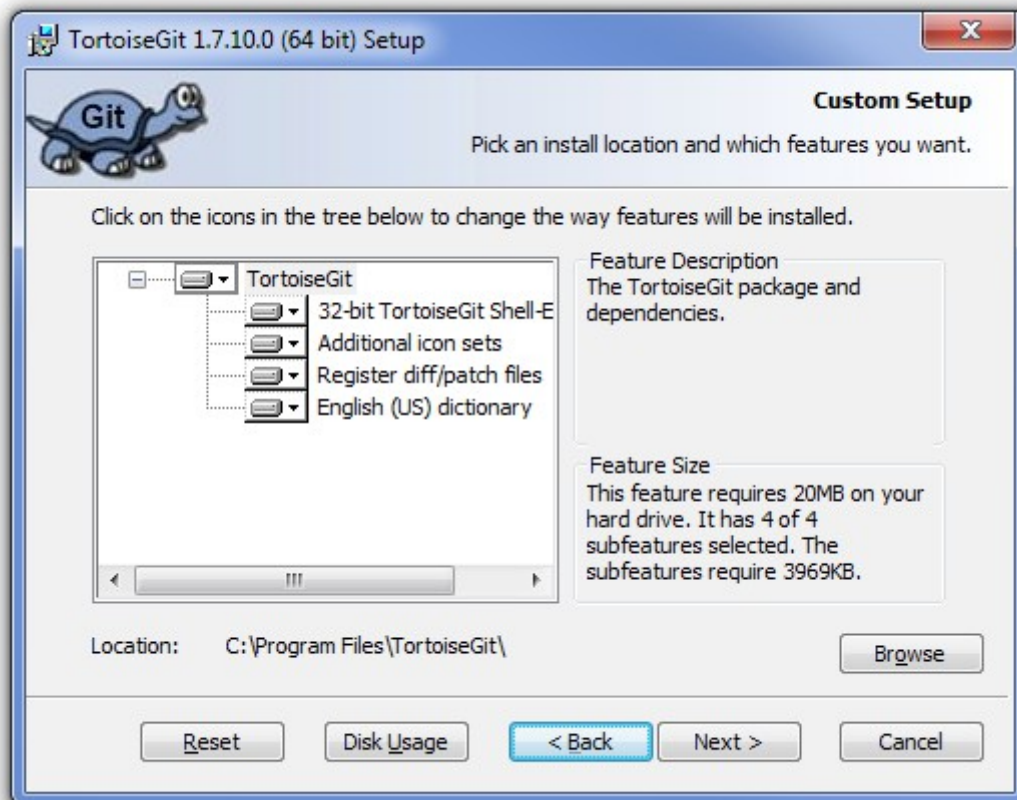


Achtet bitte darauf ob ihr ein 32 oder 64 bit System habt. In den meisten Fällen werdet ihr jedoch ein 64 bit System haben.

Und dann direkt installieren

Und einfach der Bildstrecke folgen:

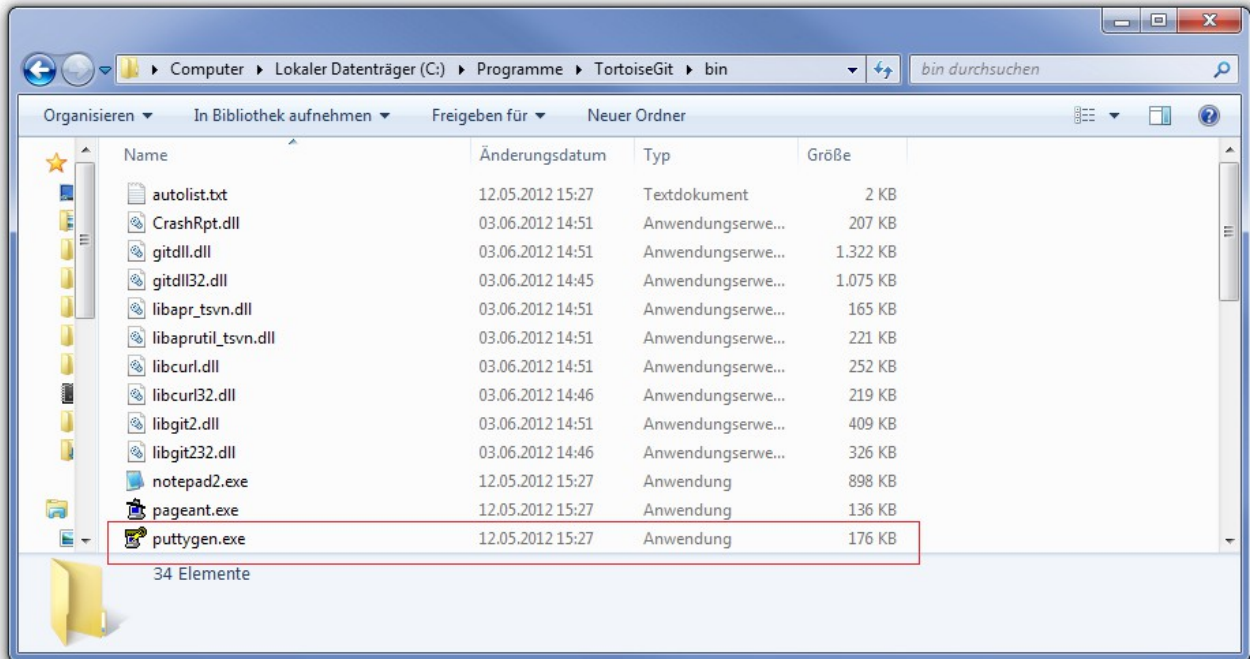




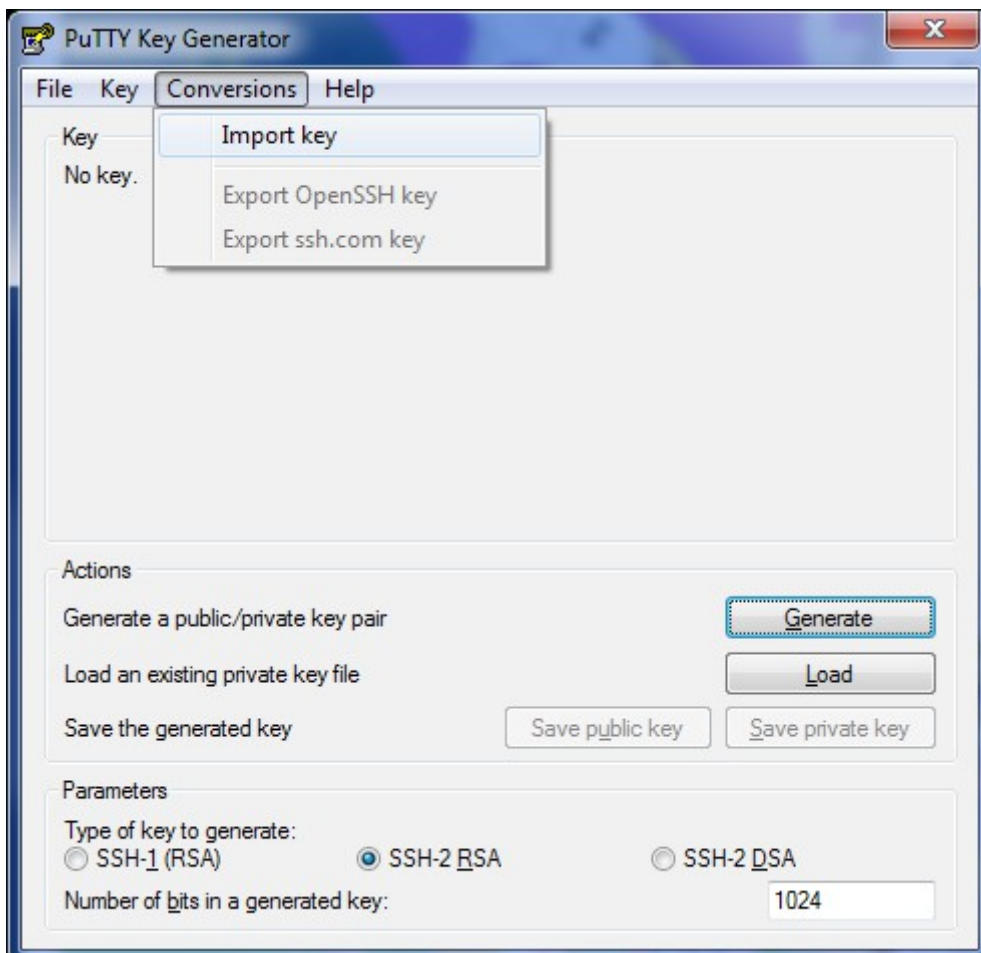
Und fertig...

.ppk Datei erstellen

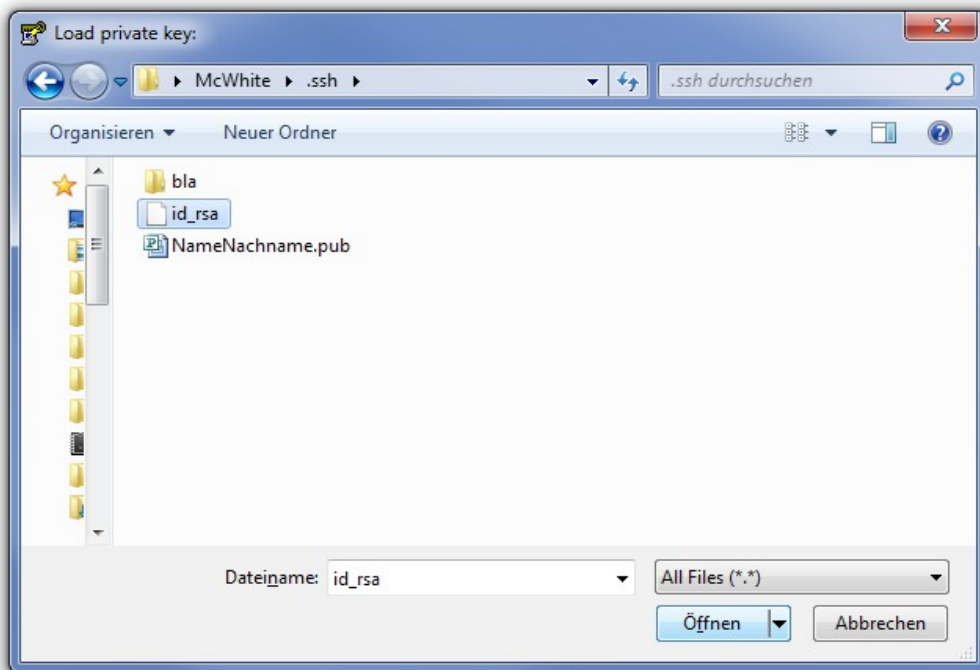
Nun geht ihr in:
C:\Program Files\TortoiseGit\bin
Und öffnet dort "puttygen.exe"



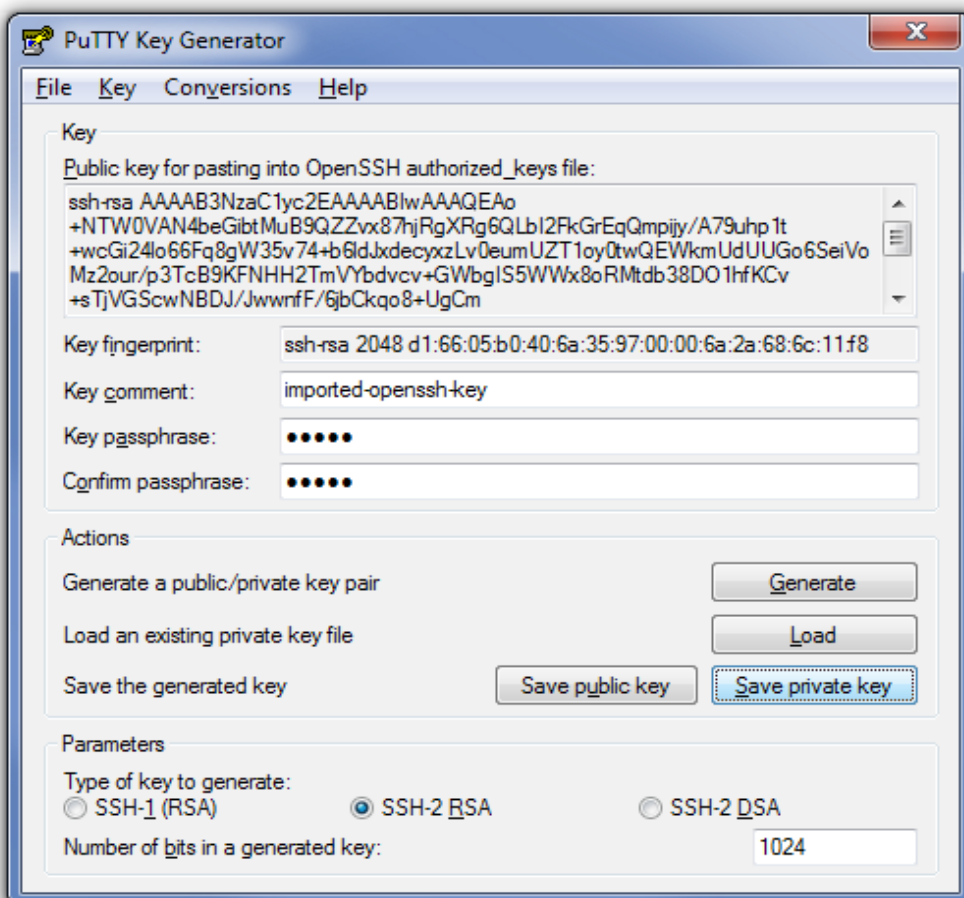
In Putty geht ihr auf Conversions -> Import key



Nun ladet ihr die andere Datei(id_rsa), also nicht die .pub



Nun müsst ihr noch einmal das Passphrase(Passwort) eingeben, dass ihr verwendet habt, um euren SSH Key zu erstellen.



Dann geht ihr auf "Save private key".

Nun speichert am besten dorthin, wo ihr das Ding immer wieder findet, der .ssh Ordner aus dem ihr die Ursprungsdatei habt, eignet sich bestens dafür. Der Name ist in diesem Fall auch egal.

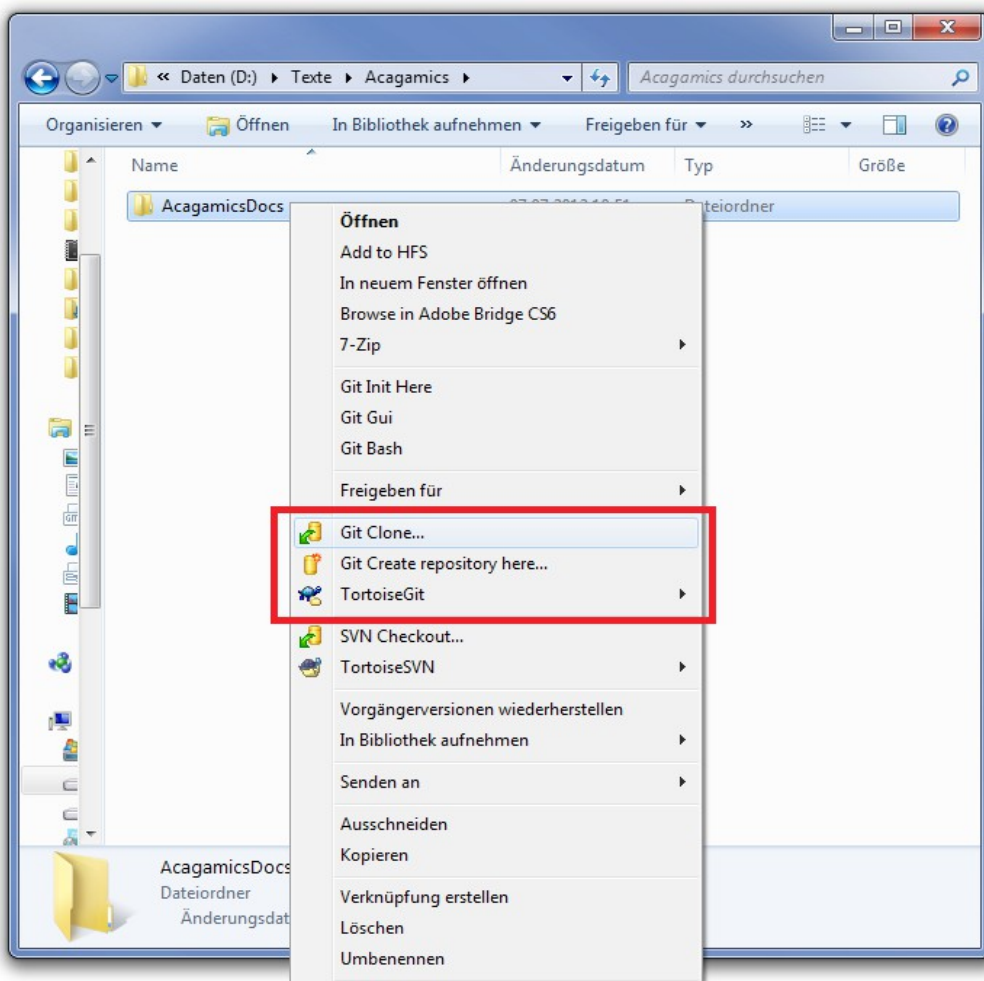
4. Tortoise Git verwenden

Repo mit eurem PC syncen

Erstellt euch irgendwo einen beliebigen Ordner in dem ihr in Zukunft die Acagamics Dokumente haben möchtet.

Nun Rechtsklickt auf den Ordner.

Dort sollte jetzt ein Punkt namens "Git Clone..." erscheinen, wenn nicht, habt ihr TortoiseGit nicht oder nicht richtig installiert.

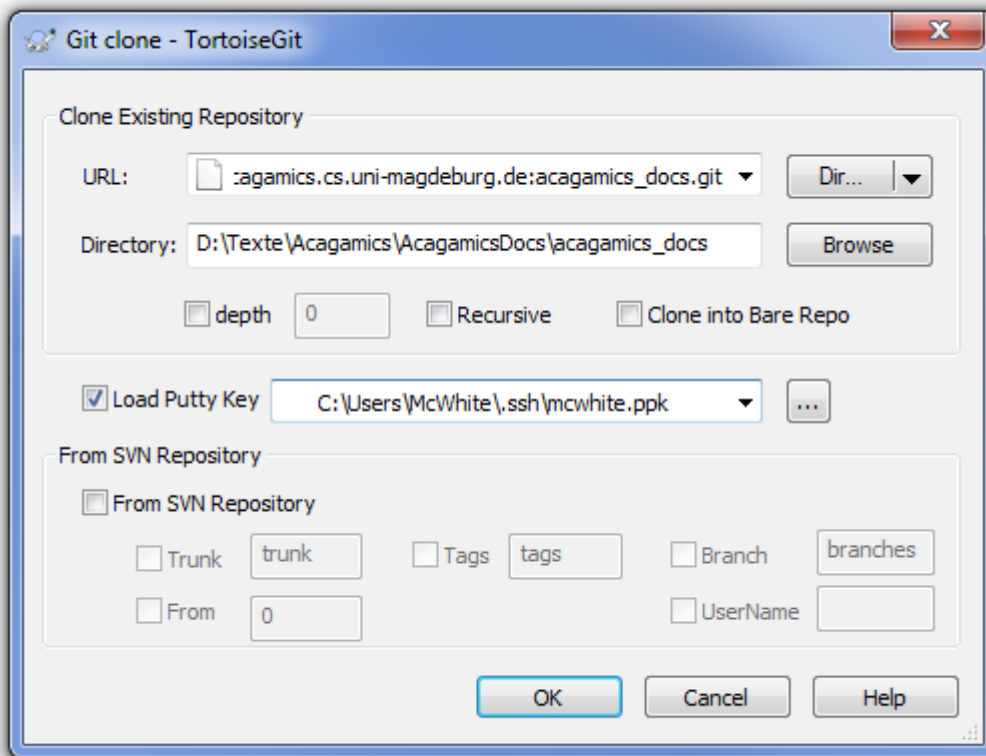


Klickt nun auf "Git Clone...". Dort müsst ihr folgendes eintragen.

URL: git@acagamics.cs.uni-magdeburg.de:acagamics_docs.git

Directory: Der Ordner sollte bereits ausgewählt sein, wenn ihr keinen Doppelordner wollt (Momentan müsste dort ..\EuerName\acagamics_docs stehen) dann entfernt einfach das "\acagamics_docs" am Ende.

Load Putty Key: Hier sollte ein häkchen sein. Drückt dann auf die "..." und sucht euren im Part 3 dieses Tutorials erstellten "WieAuchImmerIhrDasGenanntHabt.ppk" key.



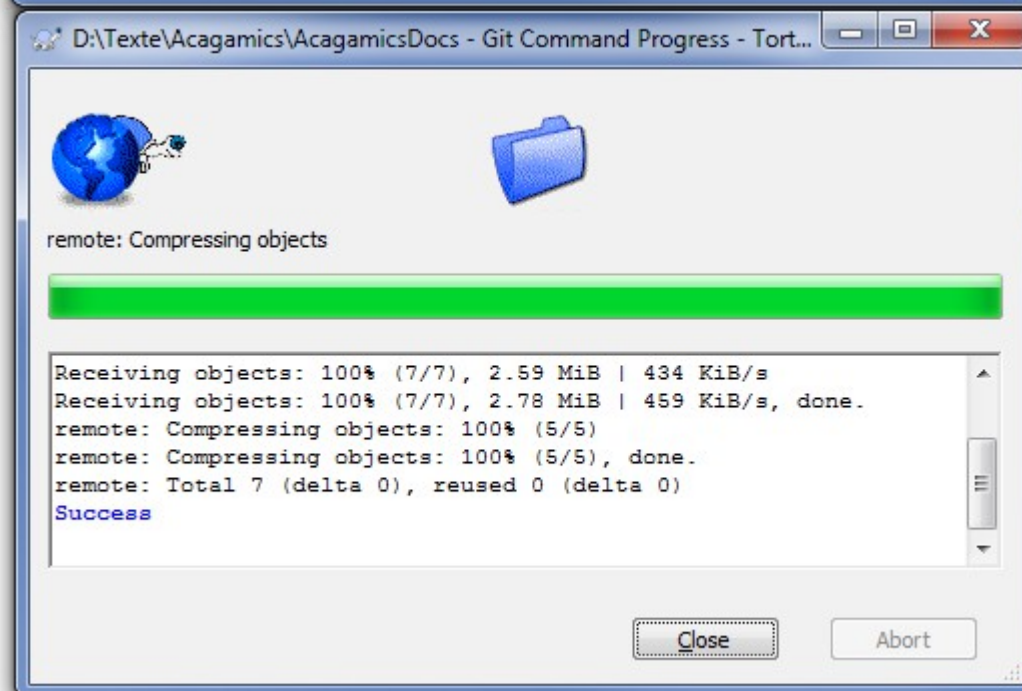
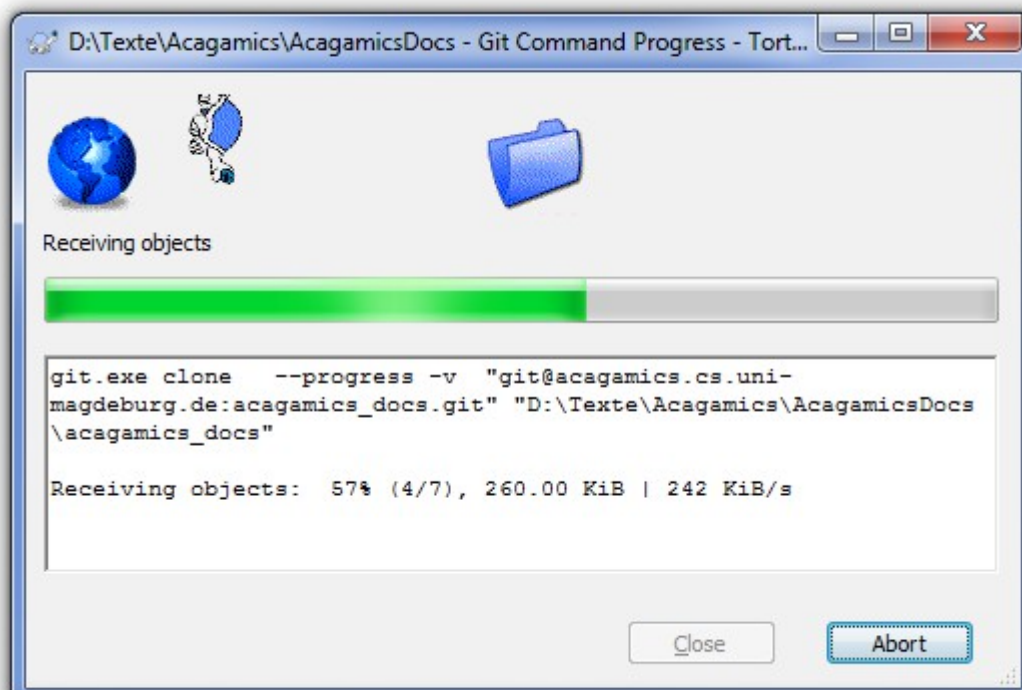
Ich würde euch empfehlen wirklich das "acagamics_docs" wegzumachen.

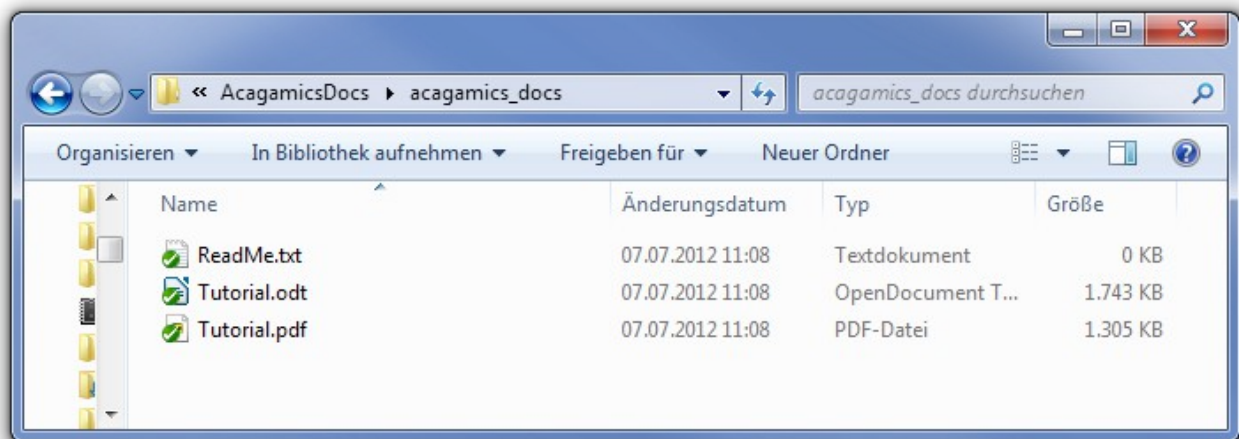
So Sollte das ganze dann ungefähr aussehen.
Drückt dann auf Ok.



Dann sollte dieses Fenster hier erscheinen. Hier gebt ihr wieder das Passwort ein, das ihr euch für den .pub und ssh Key ausgedacht habt. Das Passwort sollte auch für Zukünftige eingaben automatisch gespeichert werden.

Als nächstes werden die Dokumente geladen.



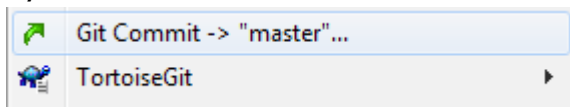


Und nun habt ihr im Ordner Dateien und Ordner mit lauter grüner Häkchen dran. Das Bild zeigt übrigens nicht den Git das jetzige Git für die Dokumente und dient nur als Veranschaulichung!

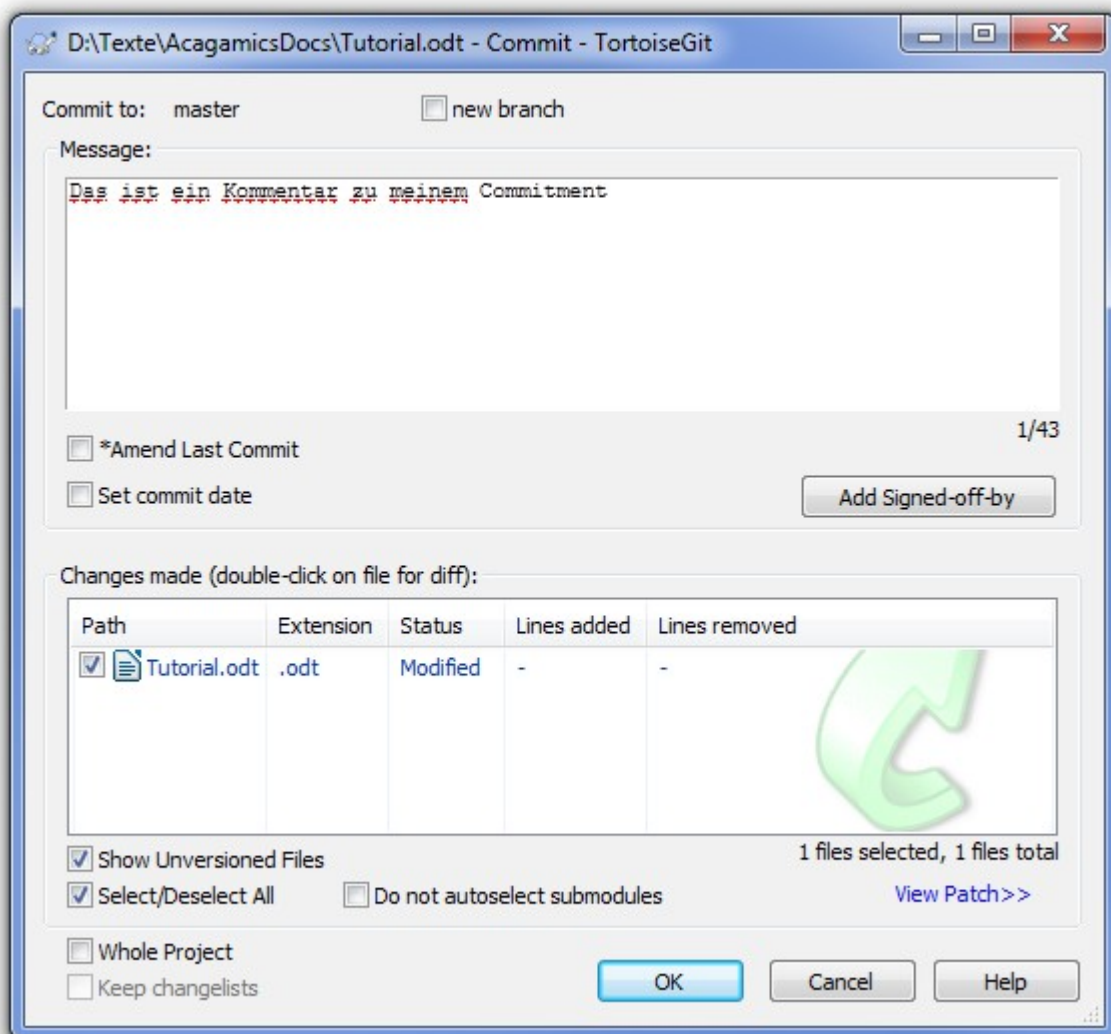
Dateien bearbeiten und commiten

Wenn ihr eine Datei oder eine Datei in einem Unterordner bearbeitet habt, erscheint ein rotes Ausrufezeichen statt dem grünen Häkchen. Dieses zeigt, dass ihr eure Version auf dem PC im Gegensatz zu der auf dem Repository verändert habt.

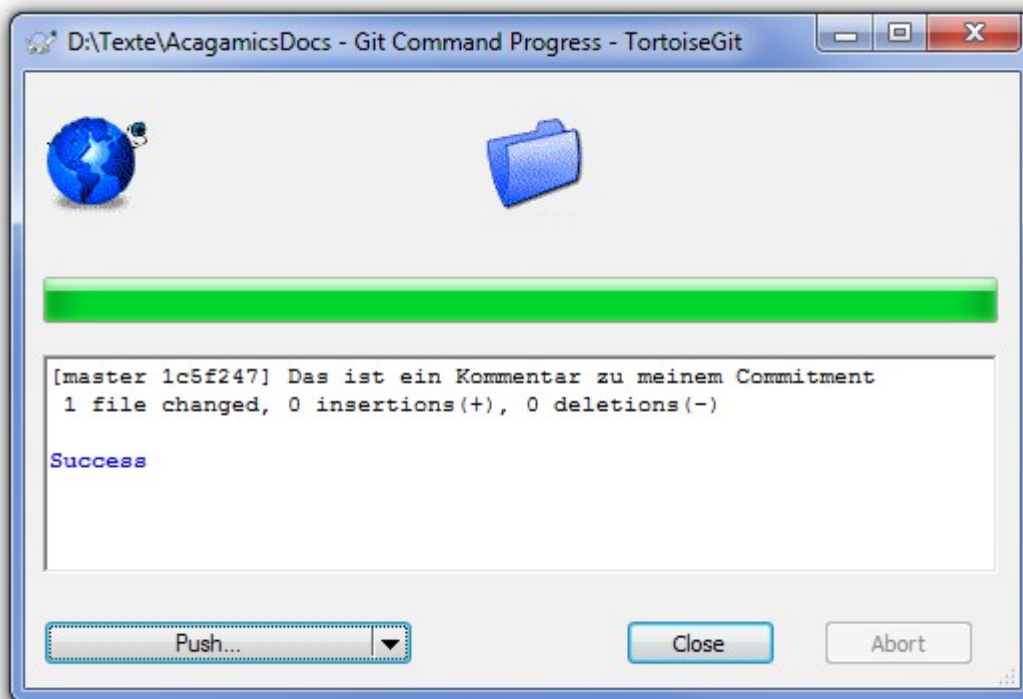
Um dies zu "beheben" rechteklickt auf die Datei/den Ordner mit dem roten Symbol und drückt auf "Git Commit"



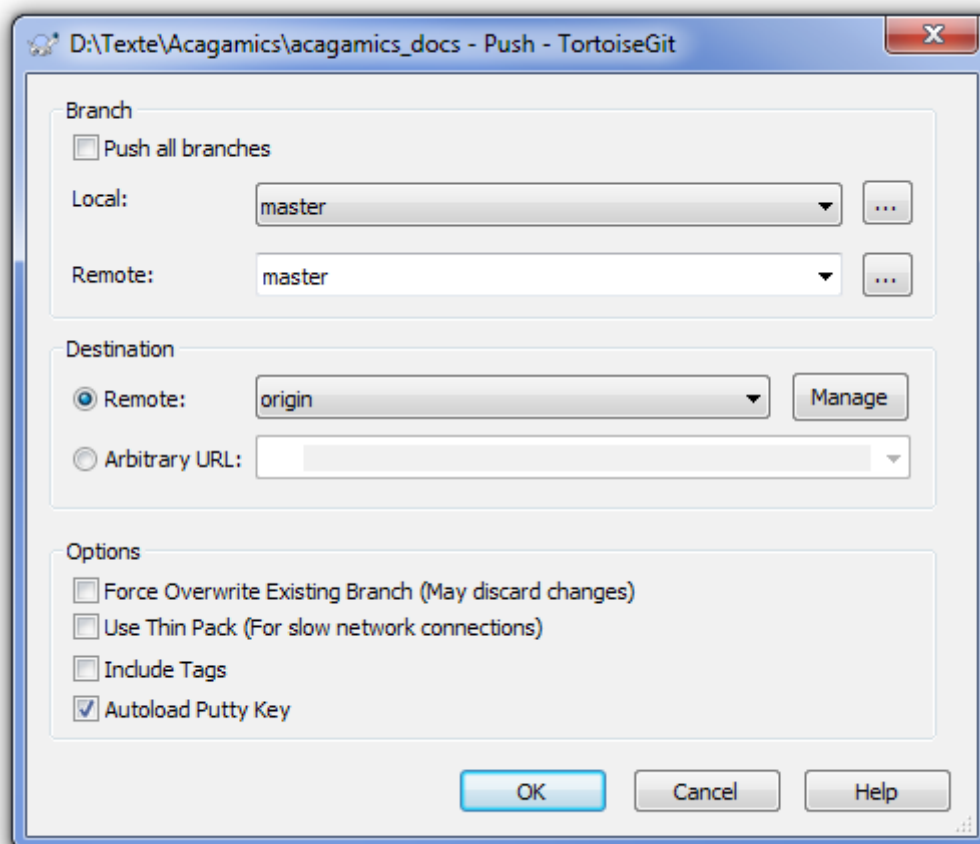
Das sollte dann so aussehen:



Wenn ihr nach einem "branch" gefragt werdet, oder die Zeile "Commit to: master" leer ist, gebt dort "master" ein



Als nächstes geht ihr Auf "Push..."



Hier wiederum einfach auf "OK" klicken.