

Git Hilfestellung

Allgemeine „Anleitung“ für die benutzung von Git in Verbindung mit Visual Studio.

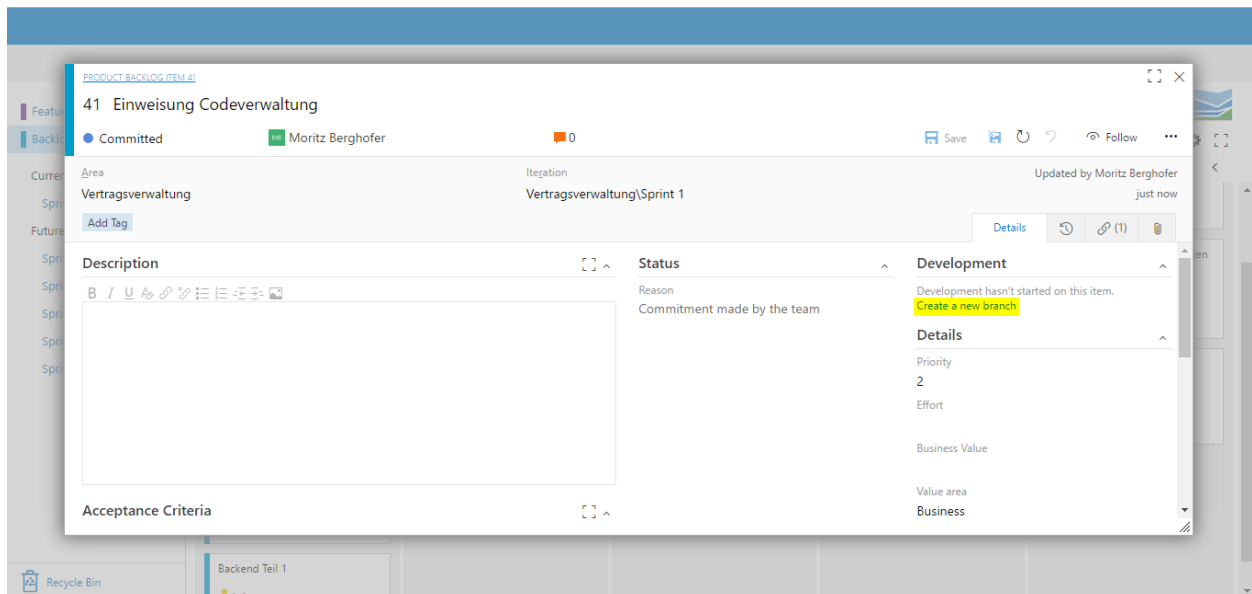
Das gewöhnliche abarbeiten eines Tasks sollte folgende Schritte beinhalten:

1. Neuer **Branch anlegen** auf dem der Code geändert wird
2. Änderung des Codes und währenddessen immer wieder **commits** auf das lokale Repository die dann auch auf den Server **gepusht** werden sollten. Falls mehrere Leute auf einen Branch arbeiten auch immer wieder die commits der anderen vom Server mit **git pull** anfordern.
3. Wenn alles fertig ist und funktioniert, **Merge Request** erstellen.
4. Merge Request wird von anderen **Überprüft** und dann wird in das Master Repository gemerged.

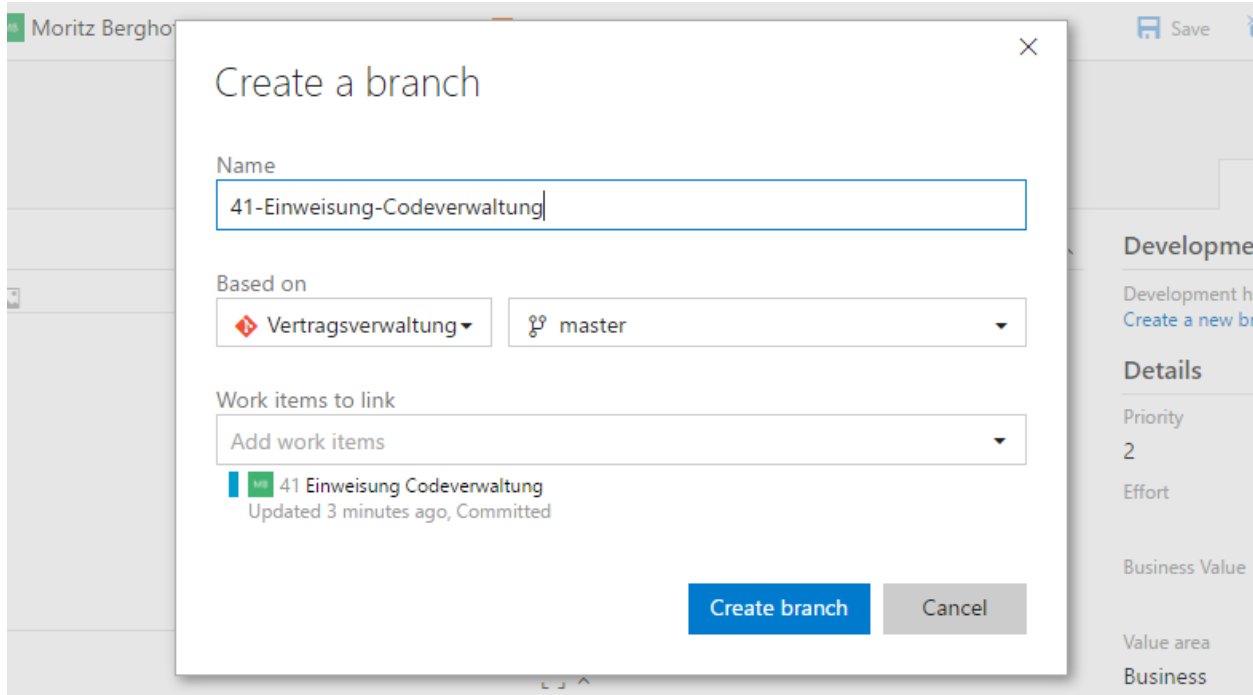
Im Folgenden wird gezeigt wie die einzelnen Schritte in Visual Studio mit den eingebauten Lösungen umgesetzt werden. Man könnte natürlich auch alles über die Konsole machen, aber so soll es zumindest einfach sein.

1. Erstellen eines neuen Branches (git checkout master ...)

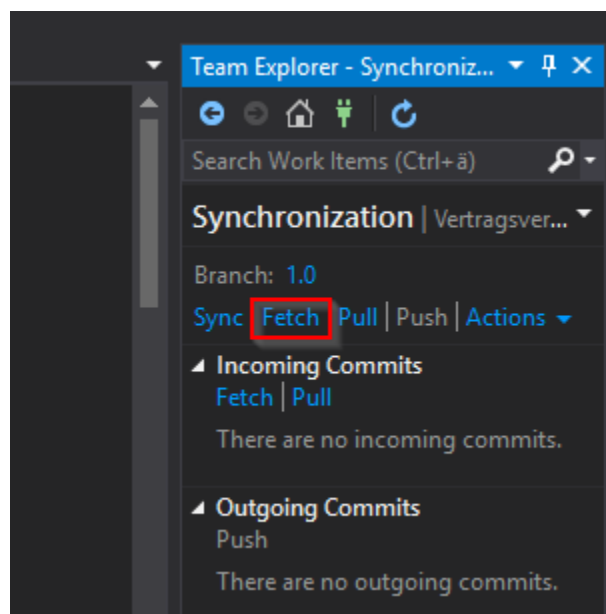
Neue Branches sollten immer Task spezifisch angelegt werden. Am besten sollte auch der Name des Branches den Task widerspiegeln. Dazu geht man ins Team Services Board und klickt dort auf seinen Task. Dort gibt es die Schaltfläche „Create a new Branch“:



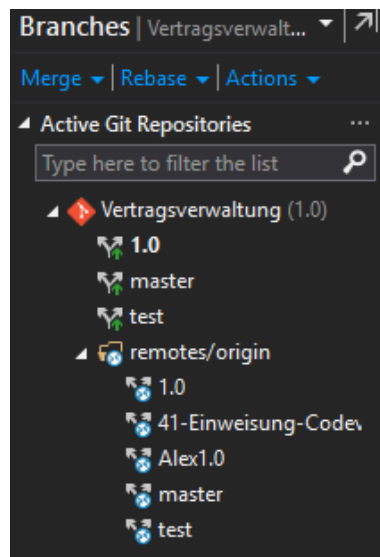
Hier dann einfach den Namen einfügen (ich habe ihn nach dem Task benannt) und von Master „abbranchen“: (Leerzeichen sind nicht erlaubt)



Danach muss noch der Branch in Visual Studio ausgewählt werden, dazu muss er erst vom Server geholt werden, dies kann entweder im Anschluss über eine Clone Schaltfläche gemacht werden oder direkt in Visual Studio, dazu muss rechts im Window auf Synchronization geklickt werden und dann ein Fetch gemacht werden:



Danach taucht der neu erstellte Branch auf Branches auf in dem Ordner remotes/origin:

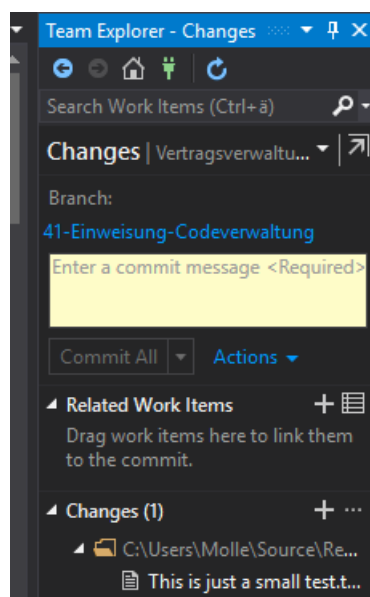


Ein Doppelklick auf den Branch lädt ihn in dein Repository.

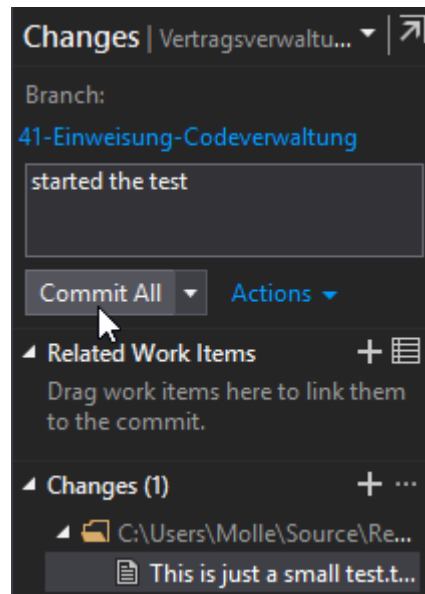
2. Bearbeiten des Codes

Wenn nun dieser neue Branch angelegt und ausgewählt ist, kann man den Code wie gewohnt bearbeiten. Dazu einfach entweder über Visual Studio die Software öffnen oder die Dateien direkt über den Windows Explorer aufrufen, sie sind nun in deinem Repository das irgendwo auf deiner Platte liegt. (Standard ist C:\Users\username\Source\Repos\Vertragsverwaltung)

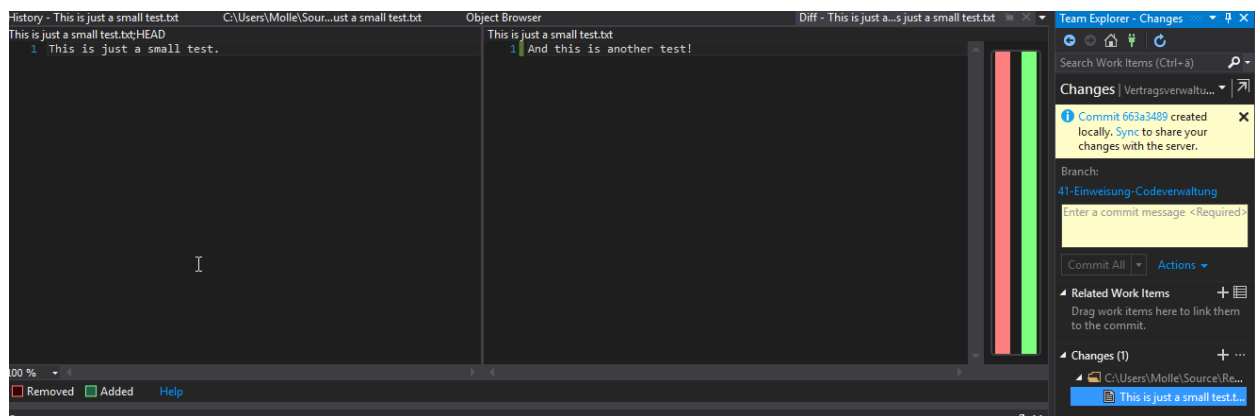
Nachdem nun der code angepasst wurde (Ich habe nur eine Textdatei erstellt, mit langweiligen Inhalt) kann man unter dem Reiter „Changes“ die geänderten Dateien sehen:



Dies kann dann erst einmal committed werden, dazu ebenfalls unter Changes eine commit Message eintragen (ist Pflicht und sollte immer die gemachten Änderungen beschreiben) und auf commit All (beziehungsweise gleich commit all and push, damit die Änderungen gleich auch auf dem Server landen) klicken:



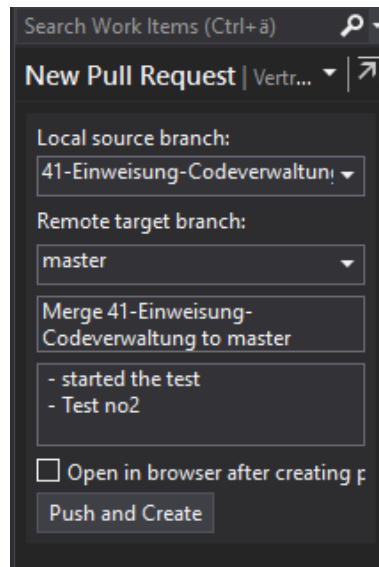
Wird nun die Datei verändert, so kann dies auch unter Changes eingesehen werden, dazu gibt es mehrere Möglichkeiten (view History, compare against unchanged) je nachdem mit was man arbeiten möchte.



Diese Änderungen können nun natürlich auch wieder committed oder eben auch gelöscht oder rückgängig gemacht werden.

3. Merge Request erstellen

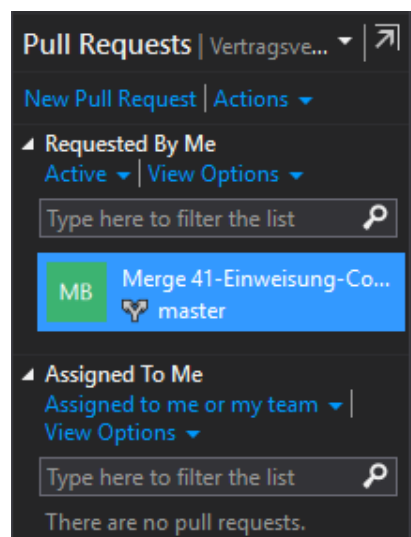
Ist man nun glücklich mit den Änderungen im Code so kann ein Merge Request (manchmal auch Pull Request genannt) erstellt werden. Dazu auf Pull Requests klicken und dann auf new Pull request.



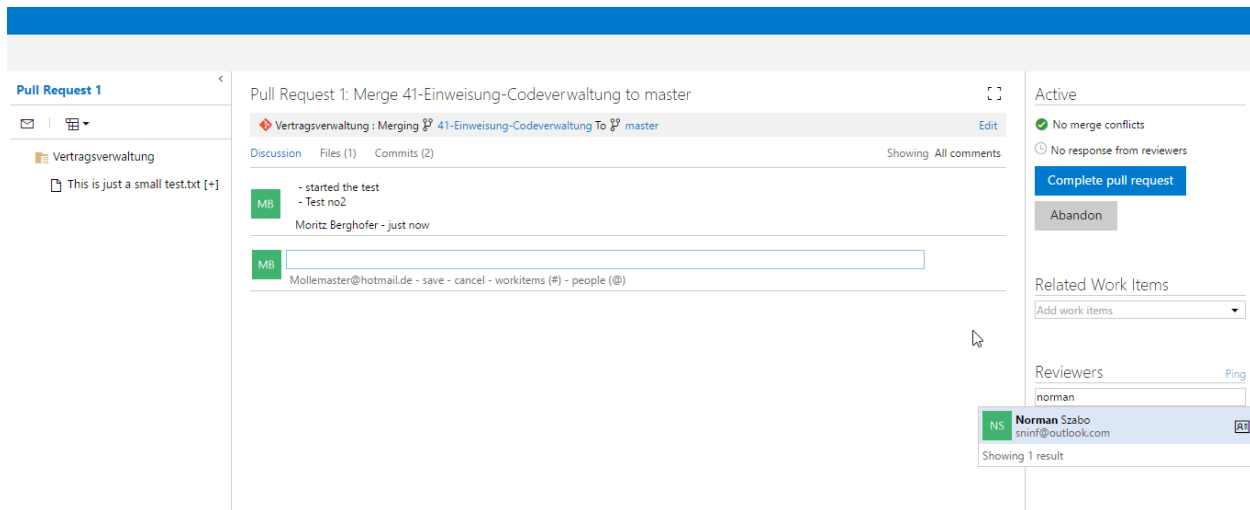
Mit einem Klick wird der Pull Request erstellt.

4. Merge (Pull) Request reviewen

Alle Pull Requests sind immer in einer Liste zu finden, dort können sie von jedem reviewed werden.



Nachdem der Pull Request erstellt wurde, können Reviewer definiert werden, die den Code anschauen müssen bevor die Branches gemerged werden:



Nachdem man nun positive Rückmeldung von den Reviewern bekommen hat, kann man den Pull Request beenden, dazu wieder den Pull Request öffnen und mit der Schaltfläche „Complete pull Request“ den Merge Request annehmen, dabei gibt es die Möglichkeit den zuvor erstellten Branch wieder zu löschen, dies sollte in der Regel auch gemacht werden. Es geht ja eh nichts verloren, da immer alle Änderungen abrufbar sind.

Ich hoffe die Anleitung hilft ein bisschen, natürlich gibt es noch viel mehr Funktionalitäten im Git aber für den Start sollte das denk ich mal reichen. Auch wenn es erst einmal umständlich wirkt, so fällt es nachdem es die ersten zwei Mal gemacht wurde gleich viel leichter und die Vorteile sieht man zwar nicht direkt, wenn man aber einen Fehler suchen muss der irgendwann gemacht wurde oder man versucht rauszufinden wann man was zerschossen hat ist man sehr glücklich darum.