

Anleitung Versionierungssystem

Falls mit dem GitLab Server der Schule gearbeitet wird folgendes ausführen:

1. Installieren und Einrichten eines VPN-Clients: [Anleitung](#)
2. Der GitLab Server steht nun unter git.htlwrn.ac.at zur Verfügung, solange die VPN Verbindung geöffnet ist.

Installation git

1. Entsprechende Konsole von folgenden Link herunterladen und installieren: [Download](#)
2. Klonen des entsprechenden Repository mit folgenden Befehl in einen Ordner deiner Wahl.
`git clone http://git.htlwrn.ac.at/i14067/SimpleQ.git`

Nun befindet man sich im Master Branch des Repository. Man sieht alle Dateien, welche im Repository liegen. Mit den nachfolgenden Befehlen kann man arbeiten.

Branches

Standardmäßig existiert der master Branch. In diesem sollte immer eine lauffähige Version des Produkts liegen. Das heißt es sollte darin nicht gearbeitet werden. Für das Arbeiten werden eigene Branches erstellt, welche vom Master abgezweigt werden und nachher wieder zusammengefügt werden.

Wichtigsten Befehle im Zusammenhang mit Branches:

<code>git branch nameDesNeuenBranches nameDesAusgangsBranches</code>	Befehl zum Erstellen eines neuen Branches von einem Ausgangsbranch
<code>git checkout nameDesBranches</code>	Befehl zum Wechseln des Branches
<code>git branch -d nameDesBranches</code>	Befehl zum Löschen des lokalen Branches

Mit dem Befehl `git branch` wird ein lokaler Branch am Rechner erstellt. Dieser wird beim ersten Push an den Server übertragen.

Benennung der Branches:

Vorschlag: <type>/<ersteller>/<beschreibung>

- <type>: bug, feature, experiment, release
- <ersteller>: Kürzel des Erstellers
- <beschreibung>: Beschreibung der Tätigkeit mit „-“ getrennt bzw. bei einem Release Branch die aktuelle Tagnummer.

Beispiel: feature/juerweb/xamarin-design

Änderungen in das Repository hochladen

Folgender Ablauf während des Programmierens:

1. Schreiben von neuen Code für einen Task. Wenn man eine Aufgabe abgeschlossen hat sollte man einen Commit erstellen. Mit diesem Commit speichert man die Änderungen im lokalen Branch ab. Diese sind aber noch nicht am Server verfügbar.
2. Wenn man nun mehrere Commits gemacht hat bzw. am Ende des Tages sollte man seine Änderungen pushen. Dies bedeutet, dass alle Commits an den Server übertragen werden und

dort gespeichert sind. Wenn man Visual Studio verwendet kann man auch direkt mit dem commit einen Push machen.

Befehle zum Commit und Push:

<code>git commit -m „Commitnachricht“ -a</code>	Befehl zum Erstellen eines Commit mit einer Commitnachricht, in welcher man kurz beschreiben sollte, was man gerade gemacht hat. Mit dem a Tag wird angegeben, dass alle Dateien, die verändert wurden mit dem commit gespeichert werden.
<code>git push</code>	Befehl zum Pushen mehrere Commits

Merge von Branches

Wenn man einen Branch mit einem anderen zusammenfüge möchte so muss man die beiden Branches mergen. Dabei behilft man sich mit Pull Requests.

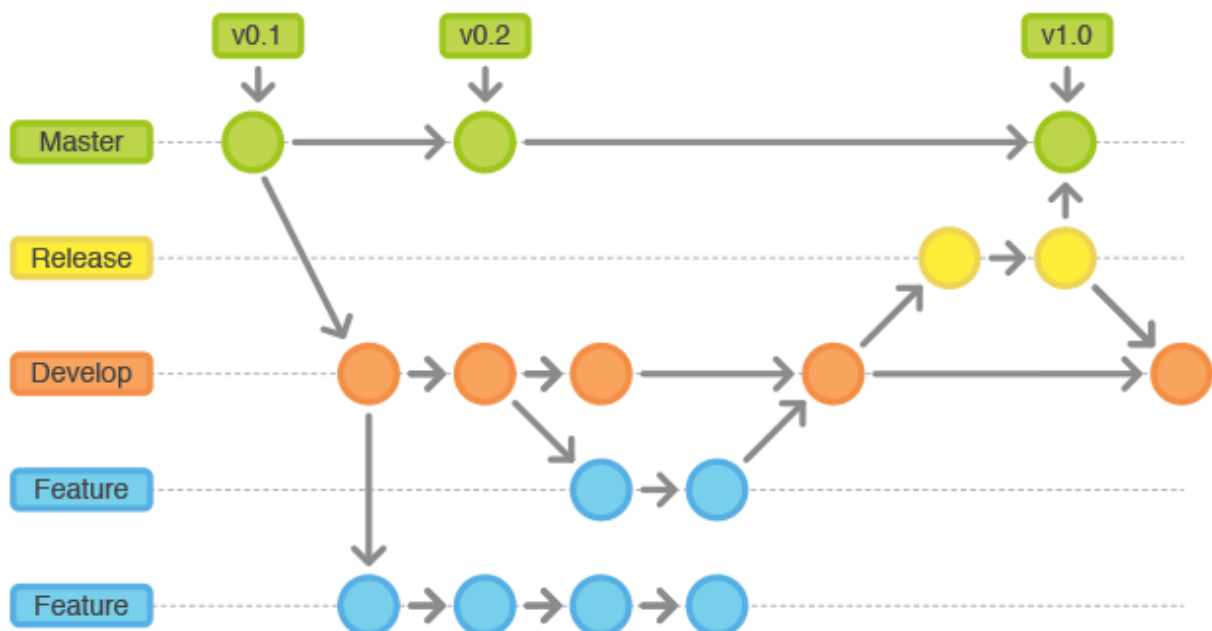
Was ist ein Pull Request?

Ein Pull Request ist ein Werkzeug um Branches einfacher mergen zu können. Dazu kann man im Webinterface des Versionierungssystem einen Request erstellen. Dies läuft folgendermaßen ab:

1. Entwickler A erstellt einen Pull Request.
2. Entwickler A behebt Merge Konflikte, falls vorhanden.
3. Entwickler A bittet um Freigabe
4. Entwickler X sieht sich die Änderungen an und gibt den Pull Request frei.

Nach einem Pull Request sollte ein neuer Feature Branch begonnen werden.

Der Gitflow-Workflow



Nach diesem Prinzip sollte entwickelt werden. Auf [dieser Seite](#) gibt es mehr Informationen zu diesem Modell.

Kurz erklärt

Wenn man im Develop Branch genügend lauffähige Features zur Verfügung hat oder man eine lauffähige Version benötigt, so wird vom Develop Branch über einen Release Branch in den Master gemergt.

Wenn man entwickeln möchte: Man erstellt sich einen neuen Feature Branch, ausgehend vom Develop Branch.

Wenn man kurz vor einem Meilenstein steht: Es wird ein neuer Release Branch vom aktuellen Develop Branch erstellt und wenn er ohne Fehler ist in den Master gemergt.