

Git Branching Konventionen

Auf dem Repository existieren exakt zwei ständige Branches: `master` und `develop`. Der `master`-Branch beinhaltet **zu jedem Zeitpunkt** eine **lauffähige** und **stabile** Version des Programmcodes.

Die eigentliche Entwicklung geschieht auf dem `develop` und davon abzweigenden Branches.

Bugfixes dürfen direkt auf dem `develop`-Branch erfolgen, so lange es sich dabei um **überschaubare** Mengen an Code-Änderungen handelt.

Feature Branches

Größere Features sollten zunächst in kleinere, gut gegliederte und separat voneinander implementierbare Einheiten getrennt werden. Für jeder dieser Einheiten wird ein neuer `feature/[feature-name]`-Branch vom `develop`-Branch abgezweigt, auf welchem die eigentliche Implementierung der jeweiligen Einheit erfolgt. Ist die Entwicklung der Einheit abgeschlossen, wird die aktuelle Version des `develop`-branch in den Feature-Branch gemergt und eine **Pull-Request** zum Merge in den `develop`-Branch erstellt. Die Pull-Request dient als Anstoß eines Code-Reviews, nach dessen erfolgreichen Abschluss die Pull-Request akzeptiert wird. Sollten Änderungen erforderlich sein, werden diese noch vor dem Merge auf dem entsprechenden `feature/`-Branch angewendet.

Review Branches

Für Korrekturen nach Reviews wird pro am Review beteiligtem Modul/Ordner ein neuer Branch erstellt. Dessen Benennung folgt dem Namensschema `review/[milestone]/[module-name]`. Meilensteine werden hierbei nach dem Schema `m[number]` benannt, also zum Beispiel `m5` oder `m6.1`. Im jeweiligen Branch beheben die Verantwortlichen die besprochenen Mängel und erstellen daraufhin eine neue Pull-Request auf den `develop`-Branch. Alle Gutachter, die diesem Modul zugeteilt waren, überprüfen, ob alle Mängel behoben wurden. Ist dies nicht der Fall werden die Verantwortlichen über Kommentare an Codezeilen darüber in Kenntnis gesetzt. Sind alle Mängel zufriedenstellend behoben wird der jeweilige `review`-Branch in den `develop`-Branch gemergt.

Feature Freeze

Vor Erreichen eines Meilensteins erfolgt ein **Feature-Freeze**. Dafür wird eine Pull-Request vom `develop`-Branch in den `master`-Branch erstellt und alle noch offenen und für den Release wichtigen Issues verlinkt (optional kann dazu auch ein Meilenstein erstellt werden) und die Arbeit an neuen Features pausiert. Stattdessen erfolgt dann die Schädlingsbekämpfung (a.k.a. Debugging) auf dem `develop`-Branch. Sind alle relevanten Issues geschlossen, wird der `develop`-Branch nach einem Review in den

`master` -Branch gemergt. Die dann aktuelle Version auf dem `master` -Branch kann dann getaggt werden. Travis kann dann den Code bauen und die Executable automatisch als Archiv an den Tag anfügen, was das Deployment erleichtern sollte.



