**Git Working-Tree/ Workflow**

Der Working Tree entspricht den Dateien, wie sie auf dem Dateisystem Ihres Arbeitsrechners liegen – wenn Sie also Dateien mit einem Editor bearbeiten, mit grep darin suchen etc., operieren Sie immer auf dem Working Tree.

Dieser Workflow definiert ein striktes Branching-Modell, das auf die nahtlose Auslieferung von Releases abzielt. Er ist sicherlich komplexer als der Feature-Branch-Workflow, bietet aber einen robusten Rahmen für das Management großer Projekte.

**BRANCHES**:

* Feature-Branches: Jedes neue Feature sollte in seinem eigenen Branch entwicklt werden, der für Backups und aus Gründen der Zusammenarbeit in die zentrale Repository gepusht werden kann. Doch statt Branches auf Basis des Master-Branchs zu erzeugen, wird der Develop-Branch als Quelle genutzt. Wenn ein Feature fertig ist, wird es zurück in diesen Develop-Branch gemergt. Dieser Workflow sieht vor, dass Features niemals direkt mit dem Master-Branch interagieren.
* Release Branches: Wenn der Develop-Branch genügend Features für ein Release enthält (oder sich ein vordefinierter Release-Termin nähert), wird vom Develop-Branch ein Release-Branch geforkt. Damit beginnt der nächste Release-Zyklus; neue Features sollten ab diesem Punkt nicht mehr hinzugefügt werden, sondern nur Bugfixes und ähnliche Release-orientierte Änderungen. Ist es zur Auslieferung bereit, wird das Release in den Master-Branch gemergt und mit einer Versionsnummer getaggt. Zusätzlich sollte es zurück in den Develop-Branch gemergt werden, der sich weiterentwickelt haben könnte, seitdem das Release initiiert wurde.
* Konventionen:  
  Branchen von: develop  
  Mergen in: master  
  Naming-Konvention: release-\* oder release/\*



* Maintenance- oder Hotfix-Branches eignen sich für das Patchen von Produktiv-Releases. Ein solcher Branch ist der einzige, der direkt vom Master geforkt wird. Sobald die Probleme gefixt sind, wird er sowohl in den Master- als auch in den Develop-Branch (oder den aktuellen Release-Branch) gemergt; der Master wird mit einer aktualisierten Versionsnummer getaggt.

Durch eine solche dedizierte Entwicklungslinie für Bugfixes kann ein Team Probleme des Produktiv-Releases beheben, ohne den Rest des Workflows zu unterbrechen oder auf den nächsten Release-Zyklus warten zu müssen. Maintenance-Branches sind sozusagen Ad-hoc-Release-Branches, die direkt mit dem Master interagieren.



Zusammenfassung

Der allgemeine Git-flow-Ablauf sieht so aus:

* Ein develop-Branch wird auf Basis des main-Branch erstellt.
* Ein release Branch wird vom develop Branch erstellt.
* Ein feature Branch wird ebenfalls vom develop Branch erstellt.
* Wenn ein feature fertig ist, wird es in den develop-Branch gemergt.
* Ist der release-Branch abgeschlossen, wird er in den develop-Branch und den main-Branch gemergt.
* Taucht ein Problem im main-Branch auf, wird ein hotfix-Branch auf Basis des main-Branch erstellt.
* Sobald der hotfix abgeschlossen ist, wird er in den develop-Branch und den main-Branch gemergt.
* 