

1. Einleitung

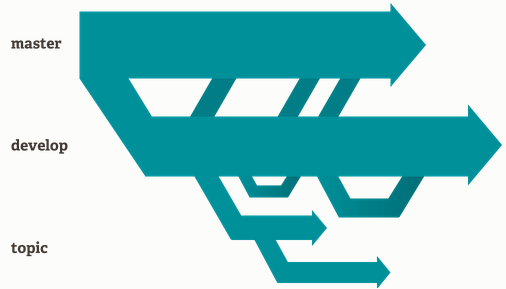
An Software Projekte arbeiten oft viele verschiedene Leute und verlieren manchmal beim Programmieren den Überblick, welche Version jetzt Online auf dem Productionserver ist oder sie wollen ein fehlerhaftes Update leicht rückgängig machen, genau diese verschiedene Leute benötigen Mechanismen, um gemeinsam an Projekten arbeiten zu können, genau diese Art von Mechanismus bezeichnet man als Versionskontrolle, in dem verschiedene Personen an den gleichen Dokumenten arbeiten zu können, denn der voll bekannten Vertreter dieser Art von Versionskontrolle ist z.B. der Software GIT und die in Zusammenspiel mit GITHUB.

GIT wirkt wie eine große Steuerung „Z“, man kann an beliebige Anzahl an schritte zurückgehen oder kann in Ruhe neue Feature entwickeln, die erst auf dem Webserver veröffentlicht werden, wenn sie komplett fertig sind.

1. About GIT



https://git-scm.com/about



1. Branching an Merging:

Heißt, dass der Prozess an dem Parallel in verschiedene Arbeit strengen gearbeitet wird, mit GIT deutlich unkomplizierter und strukturierter werden abgewickelt kann

https://git-scm.com/about/ Branching an Merging

1. Small and Fast

GIT wesentlich kompakter und deutlich schneller in dem Arbeiten von Versions Datei, abrufen ein Projekts Log dauert mit GIT unter eine Sekunde

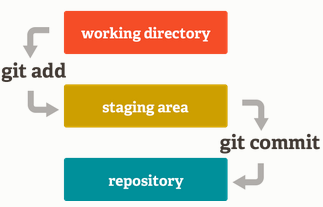
1. Distributed: Dezentrale Architektur

Jeder Nutzer von GIT hat er effektiv Kopie des Projektes auf seine Rechner liegen, das heißt kann jeder Entwickler zuerst für sich selbst arbeiten und seiner eigenen Änderung Öffentlich stellen und dann gemeinsam diese Änderungen zusammen zu führen, damit ist jeder Entwickler entkoppelt von der Abhängigkeit das zentral Repository

1. Data Assurance

Unter diesen Aspekt versteht man, das GIT sicherstellt, dass niemals Datei korrupt auf dem System liegen und keinen Inkonsequenzen stehen können.

1. Staging Area

Mit Staging Area bezeichnet GIT der Vorgang bei den Änderungen erst gesammelt werden, dann wird endlich durch den commit auf das Repository übertragen werden 

<https://git-scm.com/about/staging-area>

1. Erste Schritte in GIT

Beim GIT kann man mit zwei Möglichkeiten die Kommandos schicken, entweder mit GIT GUI oder GIT BACH, in diesem Teil wurde der GIT BACH verwendet.

Um den GIT BACH verwenden zu können, sollte man ie Funktionsweise einige Kommandos verstehen:

1. git init: mit dem Befehl git init wird ein GIT-Repository initialisiert.
2. Ein Bild, das Text enthält.

   Automatisch generierte Beschreibunggit clone: mit dem Befehl git clone wird das erstellte Projekt z.B. in GITHUB-Repository in eigene Rechner erstellt.
3. git status: mit dem befehl git status kann man den Status anschauen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Hier steht, dass kein commits möglich ist, aber es gibt ein untracked files, das heißt, es wurde eine neue datei hinzugefügt „README.md“ und die wurde commiten.

Bevor commit durchführt werden kann, muss man sie in eine stage area packen.

1. Ein Bild, das Text enthält.

   Automatisch generierte Beschreibunggit add . oder git add + Name der Aktion: mit dem Befehl kann man den file in eine stage area packen.
2. git commit: mit dem Befehl git commit und eine Message kann man die Datei in Repository übertragen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

1. git log: mit dem Befehl git log kann man sehen, welche Datei oder Änderung durchgeführt wurden.
2. git config --global user.name xxx
3. git config --global user.email xxx