Versionsverwaltung von LaTeX

Arne Brück

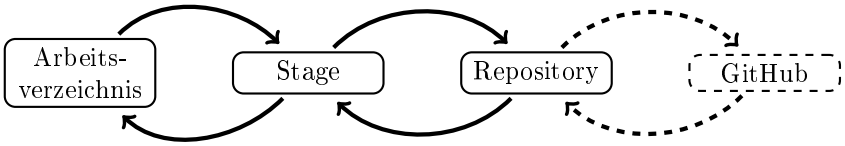
05.05.22

1 Zusammenfassung

Wer von Git hört, denkt häufig an GitHub: Dies ist der Ort, der die Hälfte der Software der Welt enthält und von dem man sich die Software kostenlos herunterladen kann.

Git ist aber unabhängig von GitHub, es ist ein Programm zur Versionsverwaltung. Es ermöglicht auf einfache Weise Sicherheitskopien des Projektes zu erstellen und zu späteren Zeitpunkten auf alle früheren Veränderungen zugreifen zu können. Dies soll anhand dieses Projektes verdeutlicht werden. Zusätzlich kann im Anschluss über GitHub das Projekt gesichert werden oder mit anderen Menschen zusammen komfortabel an diesem gearbeitet werden.

2 Grundlagen



Das **Arbeitsverzeichnis** ist das normaler Verzeichnis, in dem auf dem Computer die Dateien gespeichert werden.

Das **Repository** (Repo) ist der Inhalt des Arbeitsverzeichnis in allen gespeicherten Varianten. Wurde im Arbeitsverzeichnis beispielsweise eine Datei irrtümlicherweise gelöscht, kann ein älterer Stand des Arbeitsverzeichnisses wiederhergestellt werden, um die Datei wiederzubekommen. Hierfür muss natürlich vorher der Inhalt des Arbeitsverzeichnisses im Repo gespeichert werden. Dies erfolgt gewöhnlich immer, wenn eine Sinneinheit abgeschlossen worden ist. Hier zum Beispiel vor und nach Erstellen des Diagramms. Das Speichern im Repo wird als *Commit* bezeichnet.

Die **Stage** steht zwischen Arbeitsverzeichnis und Repository. Ein Commit ist immer ein wesentlicher Schritt, der genau dokumentiert wird und für alle anderen Menschen auf Ewigkeit sichtbar ist. Daher werden alle Änderungen zuerst

auf die Stage (Bühne) gesetzt und vor dem Commit gründlich geprüft. Idealerweise sollte das folgende Commit nicht 1 Minute später mit dem Kommentar *Diese Datei hatte ich vergessen* erfolgen.

Die Stage ist also ein Bereich in dem alles gesammelt und überprüft wird, was in der Zukunft über einen Commit im Repository registriert wird.

3 Erkärung einiger Befehle

- **git init:** Erster Befehl in einem Arbeitsverzeichnis, erstellt das noch leere Repository.
- **git add . :** Setzt das gesamte Arbeitsverzeichnis auf die Stage. Statt dem Punkt (steht für das Arbeitsverzeichnis) könnte auch der Name einer Datei angegeben werden.
- **git commit -m „Kommentar zum Commit“:** Alles auf der Stage wird im Repository mit dem Kommentar gesichert.
- **git status -s:** Gibt eine Kurzübersicht über die Dateien im Arbeitsverzeichnis, der Stage und dem Repo.

4 Aufgaben

1. Klonen Sie sich dieses Projekt mit dem Befehl: *git clone https://github.com/brueckinformatik/Git-Uebung-Latex.git* Legen Sie hierfür ein geeignetes Arbeitsverzeichnis an.
2. Erstellen Sie in Ihrem GitHub-Account ein leeres Repository mit dem Namen: Git-Uebung-Latex.git. Folgen Sie der angezeigten Anleitung und *pushen* Sie das geklonte Repository in Ihr GitHub-Account.
3. Bearbeiten Sie die folgenden Aufgaben und erstellen Sie nach jeder Aufgabe einen *Commit/Push*.
4. Schreiben Sie unter *Erklärung einiger Befehle* einen neuen Eintrag für den *git status*-Befehl ohne die Option *s*.
5. Schreiben Sie weitere Einträge für die git-Befehle: *config, clone, remote, checkout und push*.
6. Der Worstcase ist eingetreten, Sie haben aus Versehen a) den kompletten Text dieser Datei gelöscht und gespeichert, b) zusätzlich alles auf die Stage gestellt, c) zusätzlich ein Commit erstellt, d) zusätzlich einen Push durchgeführt. Erstellen Sie einen neuen Abschnitt in dem Dokument (Name z.B. Rettungsaktionen) und notieren Sie Ihre Lösungen für die Varianten.