**Thema: Dokumentenablage**

**Aufgabe 2:**

Was macht „Git“ aus?

Git ist …

* Benutzerfreundlich
* unglaublich schnell
* äußerst effizient, wenn es um große Projekte geht

**Snapshots statt Unterschiede**

* Git betrachtet Daten auf eine speziellere Art und Weise als andere Versionsverwaltungssysteme
* normalerweise werden Informationen als eine Liste von detailbasierten Änderungen gespeichert (deltabasierte Versionskontrolle)

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

* Git betrachtet seine Daten wie eine Reihe von Schnappschüssen eines Mini-Dateisystems
* nach jedem committen (gegenwärtigen Status des Projekts speichern) wird ein Abbild von allen Dateien wie sie gerade aussehen gemacht
* unveränderte Dateien werden nicht kopiert, es wird lediglich eine Verknüpfung zu der vorherigen Version der Datei angelegt
* Daten werden als Stapel von Schnappschüssen betrachtet

Ein Bild, das Tastatur enthält.

Automatisch generierte Beschreibung

**Fast jede Funktion arbeitet lokal**

* es werden keine Informationen von einem anderen Computer im Netzwerk benötigt
* Historie eines Projekts wird aus der lokalen Datenbank gelesen und ist somit ohne jede Verzögerung verfügbar
* offline zu arbeiten ist kein Problem mehr, da keine externen Server benötigt werden (Dateien werden im Nachhinein auf einen Server hochgeladen)

**Git stellt Integrität sicher**

* Git berechnet Prüfsummen von allen zu speichernden Daten und speichert diese als Referenz zusammen mit den Daten ab
* Inhalte von Dateien oder Verzeichnissen können sich nicht unbemerkt ändern
* es können keine Informationen während der Übermittlung verloren gehen oder unwissentlich beschädigte Dateien verwendet werden
* SHA-1-Hash = Mechanismus, um Prüfsummen zu erstellen
* Prüfsumme wird aus den Inhalten einer Datei oder Verzeichnisstruktur berechnet
* alles wird nach den Hash-Wert seines Inhalts gespeichert



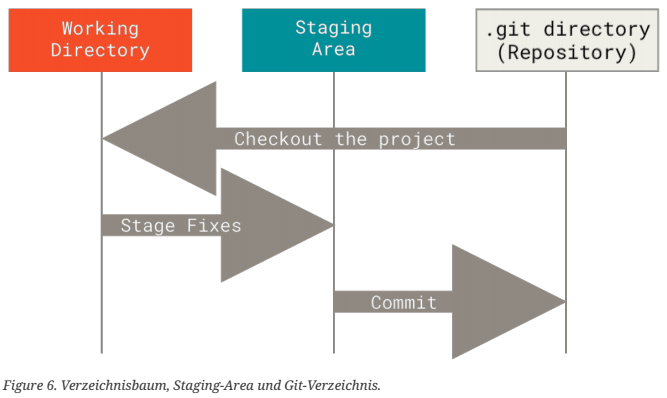
**Git fügt im Regelfall nur Daten hinzu**

* Daten werden fast immer nur hinzugefügt und somit ist es schwer das System dazu zu bewegen, irgendetwas zu tun, das nicht rückgängig zu machen ist, oder dazu, Daten in irgendeiner Form zu löschen
* Daten können nur dann verloren oder durcheinandergebracht werden, wenn sie noch nicht eingecheckt sind, sobald das aber getan wurde, ist es sehr schwierig sie wieder zu verlieren, insbesondere wenn man regelmäßig seine lokale Datenbank auf ein anderes Repository hochlädt

**Die drei Zustände**

* Git definiert drei Zustände, in denen sich eine Datei befinden kann
* Modified bedeutet, dass eine Datei geändert, aber noch nicht in die lokale Datenbank eingecheckt wurde
* Staged bedeutet, dass eine geänderte Datei in ihrem gegenwärtigen Zustand für den nächsten Commit vorgemerkt ist
* Committed bedeutet, dass die Daten sicher in der lokalen Datenbank gespeichert sind

**Hauptbereiche eines Git-Projekts**

* Verzeichnisbaum = einzelner Abschnitt einer Version eines Projekts; Dateien werden aus der komprimierten Datenbank im Git-Verzeichnis geholt und auf der Festplatte so abgelegt, damit sie verwendet oder verändert werden können
* Staging-Area = Datei, die sich im Git-Verzeichnis befindet und Informationen darüber speichert, was in den nächsten Commit einfließen soll
* Git-Verzeichnis = hier werden die Metadaten sowie die Objektdatenbank für das Projekt gespeichert; der Teil, der kopiert wird, wenn man ein Repository von einem anderen Rechner klont
* Grundlegender Git-Arbeitsablauf:
  + Dateien werden im Verzeichnisbaum geändert
  + selektiv Änderungen werden bereitgestellt, die beim nächsten Commit berücksichtigt werden sollen, wodurch nur diese Änderung in den Staging-Bereich aufgenommen wird
  + Commit wird ausgeführt, der die Dateien so übernimmt, wie sie sich in der Staging-Area befinden und diesen Snapshot dauerhaft im Git-Verzeichnis speichert