

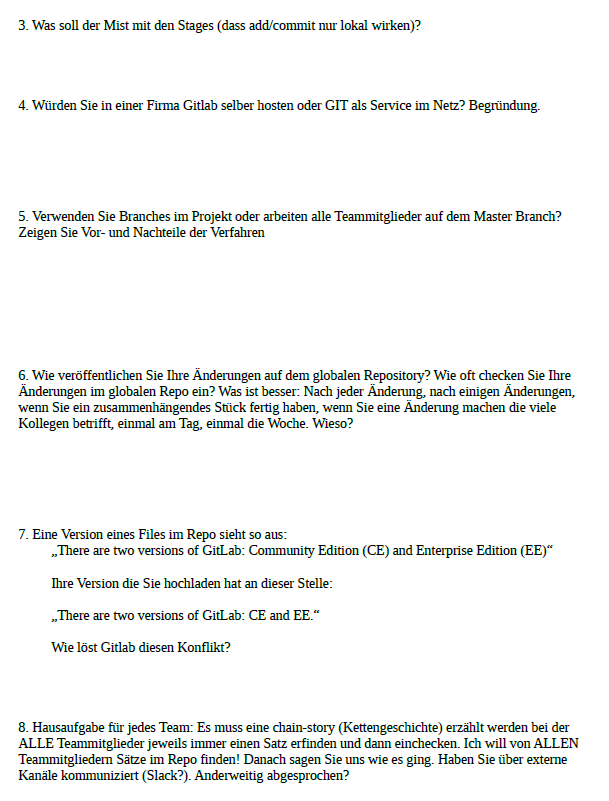
Auf Richtigkeit prüfen 😊

Kristin Behringer, 40898; Felix Schindler, 40892

* Arbeiten mit mehreren Personen
* Durch das Arbeiten mit verschiedenen Versionen des Projekts, die nachher geteilt werden, können Änderungen überwacht werden und notfalls rückgängig gemacht werden.
* Man kann Code anderen zur Verfügung stellen und aktuelle Versionen einholen
* Automatischer Code/Dokumentation kann erzeugt werden
* Fehler können einfach verfolgt werden
* - Selfhosted -> man hat die komplette Kontrolle über seine Daten –
* Open Source -> Jeder kann schauen, was mit seinen Daten passiert (Bsp. Corona-Warn-App), das Projekt selbst weiterentwickeln und Pull Requests erstellen, ... –
* GitLab CI/CD -> Pipelines ermöglichen, dass kein Fehlerhafter code auf production server gelangt, außerdem kann man routinen erstellen, die code direkt auf production server hochlädt (z. B. über rsync), Programme können dadurch auch automatisch erstellt gebuildet (Bsp. Elektron) und Releases auf git erstellt werden

Je nachdem, um was für ein Projekt Repository es sich handelt, bzw. für was man das Repo nutzt, kann man alle diese Dateien dort theoretisch ablegen. Logfiles bzw. Daten, die bei allen individuell sind, sollten nicht ins Repo.

Passwörter sollten natürlich auch nicht ins Repo.



Beide Versionen werden abgeglichen, wenn ein Konflikt auftritt, dann wird der Benutzer darauf hingewiesen und er muss ihn selbst lösen, indem man merged. Gitlab kann nicht selbstständig wissen, wie es den Konflikt löst.

Durch den push Befehl werden Änderungen global sichtbar. Vor jedem erneuten Push auf jeden Fall auf Änderungen checken, am besten schon vor der Weiterarbeit. Wenn größere Änderungen auftreten, und bei fertiggestellten Programmteilen ist es gut zu pushen. Wenn man allein an einem Programmteil arbeitet und anderen zeigen möchte, was man programmiert, eignet sich es sich auch ganz gut eine Branch zu nutzen. Richtlinien, wann ein Push Pflicht ist, gibt es nicht, man sollte jedoch keinen Code „horten“. Gepushter Code sollte laufen und vorher lokal getestet

Je nachdem, wie gut die Betreibung eines eigenen Servers bzw. evtl. Wartungsaufgaben etc. läuft, könnte man aus zur Sicherung der Daten vor Dritten Gitlab selbst hosten. Falls das zu umständlich wäre, könnte man auch einfach den Service im Netz nutzen, da dieser keine Funktionen misst.

Branches sind dann sinnvoll, wenn man eine große Programmieraufgabe hat und es unterschiedliche Gruppen für z.B. die GUI oder Logik gibt und diese unabhängig voneinander Änderungen durchführen wollen. Damit können Teilprogramme getrennt werden und z.B. erst wenn sie funktionieren, zum master gepusht werden. Sinnvoll ist dies nicht, wenn nachher jeder für sich programmiert eine andere Version besitzt und dadurch die Programme nicht mehr zusammengefügt werden können und Fehler auftreten.

Gerade bei der Zusammenarbeit mit anderen in einem Repository ist das wichtig. Da jeder verschiedene Versionen ihres Repositorys auf dem Rechner besitzt, Änderungen durchführt und dann den anderen den Code durch push zur Verfügung stellen möchte, ist es wichtig, dass zuerst lokal commited bzw. geadded wird. Unterschiedliche Versionen werden verglichen und Konflikte lassen sich leichter vermeiden.