

Cazim Ukela 40064 | Daniel Hiller 40181 | Niklas Mäcke 40060

.....  
Vorname, Name, Matrikelnummer

1. Was genau sind die Gründe um Gitlab zu verwenden?

2. Welche Daten gehören (nicht) ins Repo?

- .java Files ✓
- .xml Files ✓
- .Jason Files ✓
- Bilddateien für Gameprojekte ✓
- Musikdateien für Projekte ✓
- .project Files (Endung je nach Tool) ✗
- UML Modelle ✓
- Zeichnungen Im Prinzip nein, es kommt darauf an, was die Zeichnung beinhaltet.
- Notizen Wenn es persönliche Notizen sind dann nicht, wenn es jedoch Notizen sind, die fuer alle relevant sind, dann ja.
- Dokumentation ✓
- Configurationsfiles ✓
- Kapitel eines Buches ✗
- Eine Bachelorarbeit ✗
- Passwörter für Cloud-Services ✗
- Passwörter für lokale Services (Self-hosted) ✗
- logfiles ✗
- Messdaten vom Profiling ✗

3. Was soll der Mist mit den Stages (dass add/commit nur lokal wirken)?

So hat man die Möglichkeit fertige Änderungen zu pushen und verhindert, dass unfertige Features bei anderen Entwicklern des Projektes zu Fehlern führen.

4. Würden Sie in einer Firma Gitlab selber hosten oder GIT als Service im Netz? Begründung.

Es kommt auf die Grösse des Unternehmens an und auf deren Ansprüche. Jedoch wird für die meisten Unternehmen (keine global player) es am Sinn ergeben, wenn sie einen gehosteten Service in der Premium Funktion verwenden. Falls das Unternehmen jedoch mit sehr sensiblen Daten arbeitet und den gehosteten nicht vertraut, kann es auf eine eigene Lösung zurückgreifen.

5. Verwenden Sie Branches im Projekt oder arbeiten alle Teammitglieder auf dem Master Branch? Zeigen Sie Vor- und Nachteile der Verfahren

Es empfiehlt sich sehr mit Branches zu arbeiten und immer nach Abschluss eines Issues etc. auf den Master zu pushen, so kann man sicherstellen dass Mitglieder nicht ständig halbfertige Änderungen in ihrem Projekt haben. Ein weiterer Vorteil ist, dass das Projekt nachvollziehbarer wird für alle Beteiligten, was ggf. eine Fehlersuche erleichtert

6. Wie veröffentlichen Sie Ihre Änderungen auf dem globalen Repository? Wie oft checken Sie Ihre Änderungen im globalen Repo ein? Was ist besser: Nach jeder Änderung, nach einigen Änderungen, wenn Sie ein zusammenhängendes Stück fertig haben, wenn Sie eine Änderung machen die viele Kollegen betrifft, einmal am Tag, einmal die Woche. Wieso?

Es kommt auf die Grösse der Änderung an, prinzipiell empfiehlt es sich nachvollziehbar und übersichtlich zu committen. So das Dritte schnell nachvollziehen können, wo was gemacht wurde. Pushen sollte man spätestens nach einem fertigen Feature, so dass die Mergekonflikte übersichtlich bleiben und die Gefahr verringert wird dass doppelter Code geschrieben wird.

7. Eine Version eines Files im Repo sieht so aus:

„There are two versions of GitLab: Community Edition (CE) and Enterprise Edition (EE)“

Ihre Version die Sie hochladen hat an dieser Stelle:

„There are two versions of GitLab: CE and EE.“

Wie löst Gitlab diesen Konflikt?

Git speichert beide Änderungen in der Datei ab und fordert den Anwender auf den Konflikt zu lösen.

8. Hausaufgabe für jedes Team: Es muss eine chain-story (Kettengeschichte) erzählt werden bei der ALLE Teammitglieder jeweils immer einen Satz erfinden und dann einchecken. Ich will von ALLEN Teammitgliedern Sätze im Repo finden! Danach sagen Sie uns wie es ging. Haben Sie über externe Kanäle kommuniziert (Slack?). Anderweitig abgesprochen?