

WG Planer (lg068,lg067,lm138)

---

Vorname, Name, Matrikelnummer

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

- einfaches Teilen von Ordnern und Projekten
- alle Entwickler sind up-to-date
- leicht zu synchronisieren
- alte Versionen sind archiviert und man kann zu diesen zurückspringen
- Dokumentation des Entwicklungsprozesses

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
- Notizen
- 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)?

Es verhindert häufige Mergekonflikte

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

Selber hosten, da dadurch der unerlaubte Zugriff durch Fremde unwahrscheinlicher und besser zu kontrollieren ist.

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

Wir nutzen einen Master Branch, dadurch kommt es seltener zu großen Mergekonflikten, man erkennt schneller Disharmonien zwischen den Klassen und unser Projekt ist in der Größe auch noch überschaubar.

Durch geteilte Branches könnten wir jedoch unabhängiger von einander arbeiten und die Übersicht wäre strukturierter, wir müssten uns erst am Ende auf das Mergen konzentrieren.

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?

Vor jedem Arbeitsbeginn wird gepullt. Gepusht wird sobald ein funktionierendes zusammenhängendes Stück fertig ist. Dadurch kommt es nicht zu commits von fehlerhaften Code und jeder Entwickler kann das fertige Stück mit einbetten.

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?

Ein Unterschied besteht erst nach dem Doppelpunkt, ab da muss der Konflikt von Hand gelöst werden. Wenn der neue Commit neuer als der alte ist, dann gibts es keinen merge conflict (solange gepullt wurde zuvor). Falls nicht, `git merge`

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?

Wir haben uns über Discord angesprochen um aktiv Mergekonflikte zu testen.