

## GIT/GITLAB: Fragen und Übungen

Kay Knöpfle  
Max Knerrich  
Benjamin Mehl  
Lucca Greschner  
Vorname, Name, Matrikelnummer

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

Advantages of using Gitlab over nothing:

- Makes collaboration using git easier / possible → Provides Git Upstream
- Issue tracking
- Pull requests
- Better UX
- Providing easy access to documentation and wiki
- Providing easy to use downloads of releases

Advantages of using Gitlab over something similar:

- Open Source Software
- Self-Hosted

### 2. Welche Daten gehören (nicht) ins Repo?

Can / should be in the repository	Shouldn't be in the repository
.java Files	.project Files (Endung je nach Tool)
.xml Files	Zeichnungen
.json Files	Notizen
Bilddateien für Gameprojekte	Eine Bachelorarbeit
Musikdateien für Projekte	Kapitel eines Buches
UML Modelle	Passwörter für Cloud-Services
Dokumentation	Passwörter für lokale Services (Self-hosted)
Configurationsfiles	logfiles
	Messdaten vom Profiling

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

By not committing to upstream directly we can ensure code quality and make sure we don't push things we don't want to push yet (e.g. something with a bug to be fixed or a file we changed but is not needed for the current commit). Besides that we only have to resolve merge conflicts once rather than after every change.

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

Both ways have their distinct advantages and disadvantages. While the self-hosted approach ensures maximum independency, the SaaS approach comes with the smallest cost for maintenance and maximum ease of use.

Using a self-hosted gitlab we can also ensure that nothing leaves the company by hosting it exclusively for internal use (i.e. in the company's intranet). On the other hand, for small OSS-projects using platforms like github can boost contribution by the community as it can be found by the many users on the platform.

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

Pushing to main branch		Pushing to several branches	
Pro	Contra	Pro	Contra
Easy as in „quick and dirty“	Mistakes can have devastating consequences	Different features can be developed and tried out independently	Can become messy and confusing with big, diverse project → oftentimes requires a engaged maintainer
Can save time	Makes CI/CD harder → should be avoided	Easier to roll back planned features that turned out worse than expected	
	Makes it harder to develop multiple features simultaneously	Makes collaboration easier	
		By having to request a merge request it is ensured that everyone can take a look at new features before they get merged into main	

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?

The golden rule is not to push until you know it works – you shouldn't even commit until then. How often you commit locally before you push depends on your personal preferences and the circumstances – try to commit as often as possible to reduce data loss and ensure not to get in conflict with other collaborator's changes. Especially, if they don't trigger a merge conflict (i.e. when someone changes a file your work depends on).

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 wollen hat an dieser Stelle:

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

Wie löst Gitlab diesen Konflikt?

If the first version was on your local repo before you made your version, the first version will just get overridden.

However, when it was not on your local repo before you made your version, a merge conflict will occur. This has to be solved by you – you have to decide how to procede.

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?

It went very well. We met when we did the chain-story, so we talked to each other. Everyone knew how to solve a merge conflict.