Worksheet – Git

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

   Man hat ein übersichtliches Web-Dashboard, um alle Infos zu seinen mit git verwalteten Projekten einzusehen, darunter Visualisierungen von Commits, Merges usw.
   Außerdem können Issues eröffnet und kommentiert sowie mit Batches versehen und abgearbeitet werden.
   Auch die Zugriffsrechte sind übersichtlich dargestellt und einfach einsehbar bzw. konfigurierbar.

2. Welche Daten gehören (nicht) ins Repo?

Gehört ins Repo	**Gehört nicht ins Repo**
.java Files	UML-Modelle
.xml Files	Zeichnungen
.json Files	Notizen
Bilddateien für Gameprojekte	Kapitel eines Buchs
Musikdateien für Gameprojekte	Bachelorarbeit
Project Files	Passwörter für Cloud Services
Dokumentation	Logfiles
Config-Files	Messdaten vom Profiling

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

   Damit man nicht so viel Last auf dem Repo-Server hat (ein Push = viele gebündelte Operationen)
   Gliederung einer Menge an Änderungen in einen "Pulk", der diese zusammenfasst.
   Bei Fehler müssen nicht direkt viele kleine pushes rückgängig gemacht werden sondern man schaut nochmal über alles drüber vor dem endgültigen Push.

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

   Self-Hosted da man alle Daten bestenfalls lokal im Gebäude hat und selbst die Verantwortung für Reliability & Security trägt.

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

   Mehrere Branches:

Pro:	Übersichtlicher, besser zu managen
Contra:	Mehr Arbeitsaufwand da Zusammenführen nötig

   Alle auf dem Master:

Pro:	Kein Mergen nötig, alle Änderungen direkt bei allen übernommen
Contra:	Kann schnell unübersichtlich werden und Rollbacks betreffen oft mehrere Mitglieder

6. Wie veröffnetlichen 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?

   - Immer nach einer Working Session 1x zum Abschluss (Vorteil: Psychischer Abschluss der Arbeit)
   - Nach Abschluss eines zusammenhängenden Teils (Vorteil: Bei Aufteilung des Projekts in Sinnabschnitte gutes Weiterarbeiten durch Teammitglieder möglich)
   - Nach Änderung, die viele Kollegen betrifft (Besseres Weiterarbeiten möglich)

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 den Konflikt?

   ⇨ Merge

   Der Push der hochzuladenden Version wird rejected, da ein Konflikt auftritt. Um diesen Konflikt zu lösen, muss zuerst gepullt werden, dann der auftretende Konflikt lokal behoben werden und anschließend commited und gepusht werden.

8. Die Git-Story

   Siehe README.md in *thegitstory-llma/*