Nachdenkzettel GIT

Aufgabe 1

Gitlab allows all team members to collaborate during the whole project. In combination with version history it’s an incredibly useful tool for software development projects. Whenever a developer pushes his changes to the git repository, every other developer can easily pull the changes to his own system. With cloud repositories (like Github oder gitlab) you don't have to worry about losing the whole project due to a fire alarm or a different catastrophe. Its also a great feature to distribute the files to every developer no matter where he actually lives. The developer just have to pull oder clone the repository to his own computer and can easily contribute his work and share it with colleagues around the world. Whenever someone makes a mistake, it´s easy to restore an older version of the project because of the version control. Thanks to the branching feature, developers are able to develop in parallel to the main repository and work on other features without interrupting colleagues.

Aufgabe 2

Alle the files listed in the excercise belong into the repo except the following ones:

- UML Modelle
- Zeichnungen
- Notizen
- Kapitel eines Buches
- Eine Bachelorarbeit
- Passwörter für Cloud-Services
- Passwörter für lokale Services

Aufgabe 3

To “stage” the project (or the files you've worked on) is like caching the files you want to commit. So you can save your changes without committing them and adding them to the final product. This is very useful if you've worked your whole working time on a feature but it doesn't work as you expected or still isn't ready to use. Still, it's 7p.m and you want to go home. So you stage your work and leave.

Another use for staging could be to collect all your changes on the stage (so like storing items in a container) and then only commit once for a number of changes (like a container ship which only starts its journey once its loaded but not for one container).

With staging you simply prepare your files for a commit and check them a last time before committing them to git.

Aufgabe 4

I wouldn't see a point in hosting your own git - server albeit you companies´ size. Companies like GitHub or gitlab have specialized on git, are incredibly easy to use and cheaper then hosting your own service. Even big tech companies like Netflix, Shopify or Airbnb use GitHub. Services like GitHub are great for (remote) team collaboration, easy to setup and cheap (for smaller teams even free).

If you host your own server, you have to maintain them which costs energy and resources (time, money and labour). Maybe you could be more flexible with your own VCS, but because git is open source you would end with a pretty similar product like GitHub.

I wouldn't recommend setting up your own VCS and sticking to gitlab or GitHub because you don't have to reinvent the wheel for your own VCS. It would make more sense to save your resources which you can then put into your product.

Aufgabe 5

It’s always useful to use branches and add the branch to the master branch after achieving a “checkpoint” or an issue. Now you can be sure that everyone is up to date. Furthermore, many programmers can work on different branches without disturbing each other. But when merging the branches it might be stressful and annoying because

of merge conflicts. But still, the project might be more comprehensible for programmers and it facilitates the error-search

Aufgabe 6

With `push`. It’s reasonable to push quite often in order to avoid version conflicts. “Release often, release early”.

Aufgabe 7

Git saves both changes and asks the user to solve the merge conflict

Aufgabe 8

It worked fine, we communicated through our whatsapp group. You can see our result on our git page.