

Git & the project

Short guide of everything you need to consider when using Git during the project.

Where to host the central repository

This is completely up to you. If you want, you can host it locally at school or in a private location. However, the easiest solution is probably to use either Google Code or GitHub. These services also offer some extra features, such as an issue tracker and a wiki. *Regardless of the hosting you choose, it is important that the supervisor can access your repository during the course.* This is needed not only for grading but also in preparation for each supervised meeting.

How to work with Git

Any commits done to the repository should be small and *only deal with what you describe in the commit*. Never commit enormous changes in one single commit. A well-structured commit log gives us a much better impression and makes it easier for us to follow your work. Two very positive aspects during grading. For this reason, it is important that you plan your work carefully. Using Git-gui when committing can often help in this regard, as it lets you select specific files or specific sections in files for each commit.

Whenever you do any restructuring of files in your repository, always do that in a commit that does only that. Never mix actual work and movement of files in the same commit. Furthermore, you should never remove a file and then re-add it in a later commit. This can, in some cases, confuse Git and mess up your history.

Your development branch (where you do all your work) should always be on master.

Commit messages

Commit messages should be informative and describe the work done in the commit. Also, the commit messages of Git repositories should follow this convention:

- > Short summary of the changes.
- >
- > Whenever needed, a more in-depth description of the changes
- > ... which can go on for as many lines as you want.

Each commit line should never be more than 75 characters wide. If you use Git-gui when committing, this is actually the default width configured for commit messages.

Meeting agenda tags

Whenever you have a group meeting, you should add your meeting agenda to the repository, in text format, using an annotated git tag. This enables us to track your meetings, read them, and quickly see at which point of development each meeting happened. You should also commit your meeting agenda and put it somewhere in the file structure of the repository.