GIT training

Syllabus

The beginner training will focus on working with a local repository, including branching and merging. You will learn, that working with git can give you the safety, to be able to return always to any state which was previously commited.

The intermediate training will introduce the remote repositories, and the different strategies to work with them. For different organisational structure there is matching method. We will learn how to use branches if there are only a handful developers sharing a single repo, and also the recommendations for geographically distributed, larger organisations with different access right needs.

There is an optional "interactive rebase" workshop, which focuses on how to work with long living branches, and how to solve merge conflicts. All kind of modification is possible: changing a commit (message or content) removing unwanted changes/files. You can even change the order of commits, or join multiple commits into a single one. Anything is possible.

1. Begginer - 1 x 4 hours

1.1 The tale of the 3 trees: repo/stage/workdir

1.2 Create a repo: `git init`

1.3 Your best friend: `git status`

1.4 First file about to save: `git add`

1.5 Lets make it safe: `git commit`

1.6 What are the plumbing commands?

1.7 Lets take a look into the .git dir: everithing is a file

1.8 Lets look into the stage: `git ls-files`

1.9 File modification

1.10 What did we do: `git diff`

1.11 Lets commit some more

1.12 How about a new file

1.13 Now we have a history: `git log`

1.14 Learn abour references: HEAD

1.15 How is your dady?

1.16 Back to the future: `git rest`

1.17 Soft/Mixed/Hard reset, what are they good for.

1.18 Lets label everything: `git tag`

1.19 Bug fixing process: `git branch`

1.20 Jumping between branches: `git checkout`

1.21 Get your changes into a different branche: `git merge`

1.22 What is git spagetthi.

1.23 A different type of merge: `git rebase`

1.24 Pros and cons of merge/rebase

2. Intermediate - 1 x 4 hours

2.1 Remote repo: `git clone`

2.2 What are the supported protocols: file/http/https/ssh/git

2.3 Remore repo = alias + url

2.4 Long branch names, what are those strange '/' characters: `origin/master`

2.5 How to get your local repo up to date? No, its not `git pull`

2.6 The right way: `git fetch`

2.7 pull = fetch + merge

2.8 How to work on multiple branches

2.9 Merging in a team? Who is the quickest?

2.10 The slowest has the most task?!

2.11 Merge or Rebase the final countdown

2.12 Lets do both ways, and check the differences

2.13 What is forking?

2.14 How to send you changes to OSS projects?

2.15 Branching strategies: gitflow or github-flow

3. Customizable optional material depending on knowledge level and client needs 1x4 hours

2018.03.23.