

Agenda

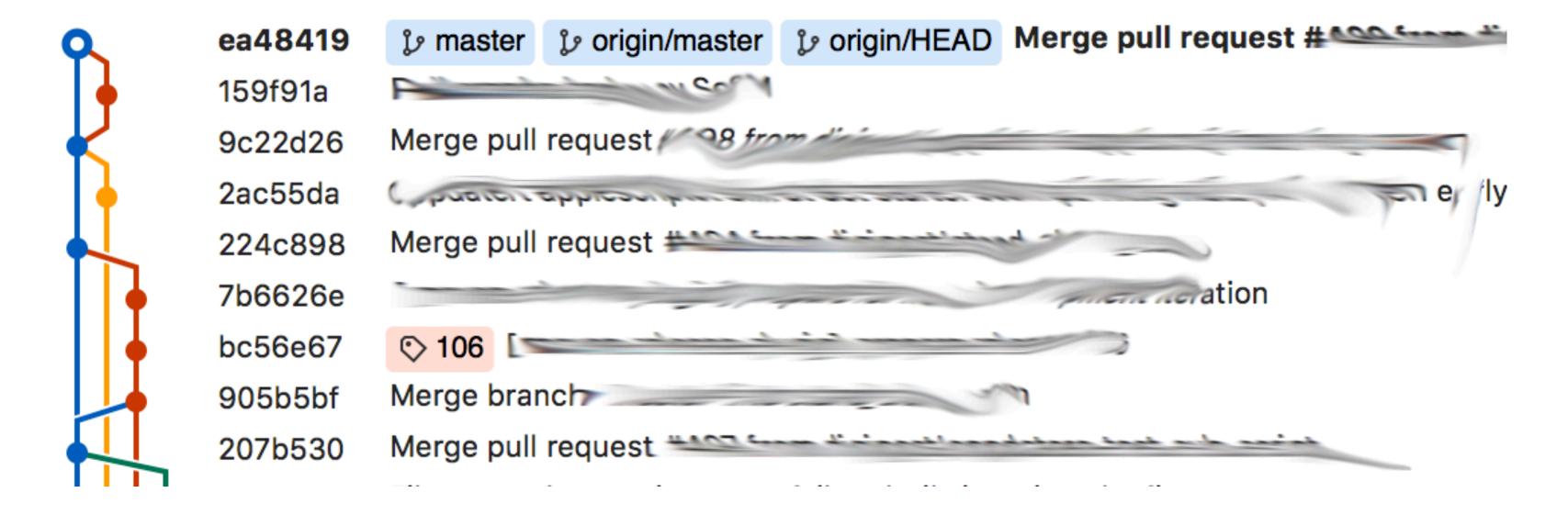
- 1. Håndsopprekning
 - 2. Introduksjon
 - 3. Motivasjon
 - 4. Oppgaver
 - 5. Verktøy
 - 6. Oppsummering

Deres erfaringer

- 1. Brukt git
- 2. Alene, som «backup»
- 3. Sammen med andre, basic: branch, merge
- 4. Avansert: rebase, interaktiv, branching-strategier, stash, squash, reflog

Litt om Git og versjonskontroll

HVa er versjonskontroll?



git status

git add <fil>

git commit

git push

HVOrfor bruker vi det?

1. Historikk

2. Arbeidsflyt under utvikling

- 1. Brancher
- 2. Diffing og commits
 - 3. Pull requests

3. Arbeidsflyt i et team

- 1. Flere features i parallell
- 2. Egne brancher som representerer «prod»
- 3. Rulle tilbake til tidligere versjoner ved bugs

opgaver

Lag bruker på github.com

github.com/sindrebn/git-kurs

Verktøy

IDEer

- → IntelliJ
- → Visual Studio

Frittstående programmer

- → SourceTree
 - → GitUp

Config

```
[alias]
   st
            = status -s
   ci
            = commit
            = checkout
   CO
             = log --graph --abbrev-commit --date=relative
   lg
                   --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset'
   logline = log --pretty=\"%C(auto)%h %ar %Cblue%s%Creset%C(auto) %d %an\"
             = shortlog -sn --no-merges
   rank
             = whatchanged --oneline
   unstage = reset HEAD --
   uncommit = reset --soft HEAD^
   praise = blame
            = submodule update --recursive
    SU
             = diff -w --ignore-space-at-eol
    diw
[pager]
    diff = diff-so-fancy | less --tabs=4 -RFX --pattern='^(added|deleted|modified): '
    show = diff-so-fancy | less --tabs=4 -RFX --pattern='^(added|deleted|modified): '
[push]
   default = current
   followTags = true
[fetch]
   prune = true
```

Oppsummering 5 råd på veien

Oppsummering

- 1. Les manualen
- 2. Ikke gå bananas
- 3. Bruk pull requests
- 4. Åpne PR tidlig, lukk den raskt
 - 5. Er du i tvil spør en venn



Sindre Nordbø, @sindrebn