Versiebeheer

Aka: Git voor dummies

AP HOGESCHOOL ANTWERPEN

AP.BE

Wat is goed versiebeheer

- Een manier om wijzigingen te tracken
- Een manier om samen te werken
- Een manier om verschillende versies samen te voegen
- Een manier om verantwoordelijkheid te tracken

Hoe doe je aan versiebeheer

- Veel kopieën (tekstV1.0, tekstV2.0, tekstFin,…)
  - Onoverzichtelijk
  - Vereist discipline in naamgeving
- Een versiebeheersysteem (git)
  - Automatiseert versiebeheer
  - Accountability
  - Distributed collaboration
  - Reversability

GIT


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

GIT: geschiedenis

- Linus Torvalds (maker of Linux) = sociaal incapabel
  - Wilt samenwerken maar niet met mensen spreken

  ==> Bedenkt GIT om met mensen samen te werken

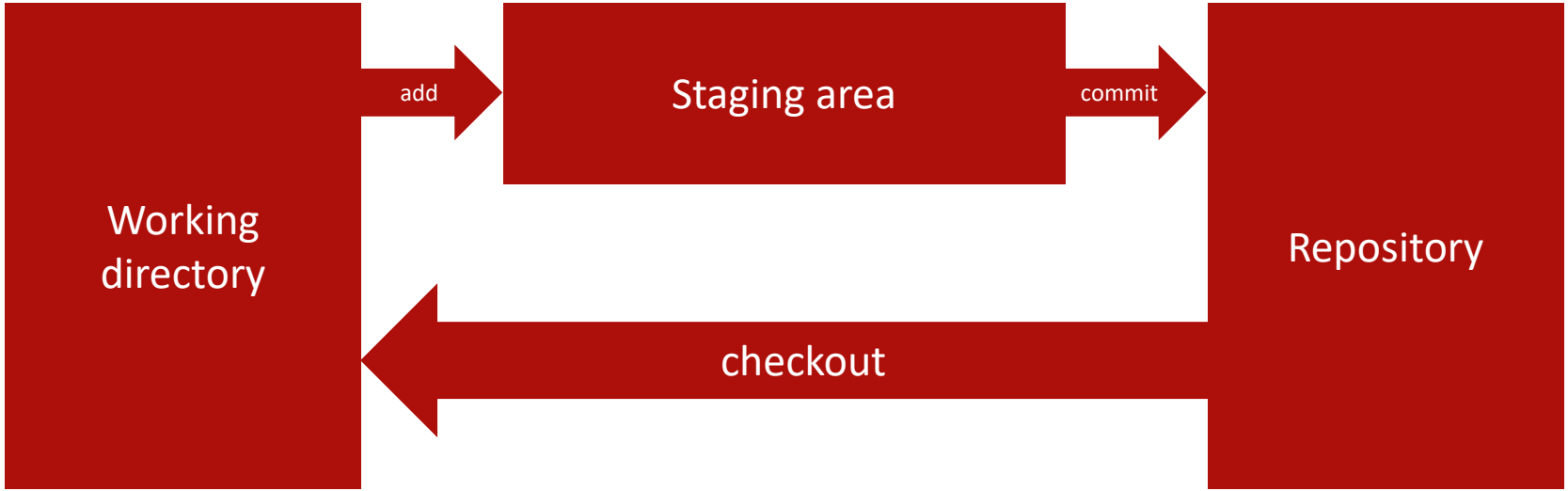
GIT != github

- Git is een versiebeheersysteem
- Github is een hosting provider die met git werkt
  - Alternatieven beschikbaar (Gitlab,…)

GIT werkingsprincipe

- Working directory
  - Actieve werkmap met bestanden die je kan bewerken
- Staging area
  - Verzamelplaats voor wijzigingen die je wenst te bewaren
- Repository
  - Permanent bewaarde wijzigingen georganiseerd in branches

GIT werkingsprincipe



GIT basic commands

git init

Maak een nieuwe git repo aan in de huidige map

GIT basic commands

git clone <link to repo>

Kopieert een repository naar een lokale map

GIT basic commands

git add <bestand>

Voegt de wijzigingen aan een bestand toe aan de staging area.

Gebruik "." om alle wijzigingen aan alle bestanden toe te voegen.

GIT basic commands

git add <bestand>

Voegt de wijzigingen aan een bestand toe aan de staging area.

Gebruik "." om alle wijzigingen aan alle bestanden toe te voegen.

GIT basic commands

git commit -m "<commit message>"

Voegt de wijzigingen die in de staging area staan toe aan een commit met de gegeven commit message. Deze commit wordt dan in de repository geplaatst

GIT basic commands

git log

Toont alle commits die tot nu toe gemaakt zijn met de bijhorende commit message

GIT branches

- Verschillende paden van wijzigingen
- Paden kunnen samen komen in een merge
- Conflicterende wijzigingen ==> merge conflict (zie volgende les)
- Best nieuwe wijzigingen op een apparte branch maken

GIT basic commands

git branch

Toont alle beschikbare branches

GIT basic commands

git checkout <branch name>

Plaats de laatste commit van de opgegeven branch in je working directory.

GIT basic commands

git checkout <commit ID>

Plaats de opgegeven commit in je working directory.

Opgelet!: aangezien je aan een commit uit het verleden gaat sleutelen zal je een nieuwe branch moeten maken om deze wijzigingen in onder te brengen.

Gebruik "git checkout <branch name>" om terug naar het einde van je branch te gaan

GIT basic commands

git checkout -b <branch name>

Maak een nieuwe branch vanuit de commit waarin je working directory zich nu bevind.

GIT basic commands

git log --graph --all

Toon alle commits op alle branches in een boomstructuur.

GIT basic commands

git merge <branch name>

Incorporeert alle veranderingen op de aangeven branch in de huidige branch. Kan een merge conflict opleveren indien 2 identieke stukken code conflicterende wijzigingen hebben gekregen. (zie volgende les voor oplossing)

Remote intermezzo

Something about ssh keys...

AP HOGESCHOOL ANTWERPEN

AP.BE

Simpeler remote werken

- Stel je ssh key in op github/gitlab/bitbucket/...
- Gebruik ssh voor al je verbindingen!
- Typ nooit nog een wachtwoord te veel in

GIT remotes

Makes it worth your time

AP HOGESCHOOL ANTWERPEN

AP.BE

GIT remotes

- Branches op afstand
- Andere computer ==> asynchroon
- Remote branches kunnen 'gekoppeld' zijn. Als er zich toch wijzigingen op de remote branch bevinden en ook wijzigingen op de lokale branch sinds de laatste sync zijn deze 2 branches verschillend en zal je ze dus moeten mergen.

GIT remote commands

git remote add origin <link to repo>

Voegt een remote git repo toe onder de naam origin

GIT remote commands

git remote remove origin

Verwijdert de remote locatie genaamd origin.

GIT remote commands

git pull [<remote repo> <remote branch>]

Gaat kijken of er remote changes zijn en voegt deze dan toe aan de lokale repo. Voert achterliggend een merge uit met de remote branch

GIT remote commands

git push

Plaatst alle nieuwe wijzigingen in de lokale repo in de remote repo

Let op!: als de branch waarop je je bevind nog niet bestaat op de remote repo zal git je vragen om de optie "--set-upstream <remote-repo> <remote-branch>" te gebruiken. Dit moet je doen anders zullen je wijzigingen niet in de remote repo verschijnen.

Opdracht

Blah blah is over, time for boom boom!

AP HOGESCHOOL ANTWERPEN

AP.BE

GIT opdracht 1

1. Maak een git repo aan op je eigen computer
2. Plaats in deze repo een bestand met de naam "name.txt"
3. Voeg het bestand toe aan je staging area en commit de wijzigingen. Geef als commit message "initial commit"
4. Voeg je naam toe aan het bestand, en voer stap 3 opnieuw uit met nu als commit message "Add name to name file"
5. Maak een nieuwe branch aan met de naam "number"
6. Maak in deze nieuwe branch een commit waarin je je studentennummer toevoegt aan het name.txt bestand. Geef deze commit de commit message "Add student number to name file"
7. Ga nu terug naar de main branch en maak een commit aan waarbij je een bestand toevoegd genaamd "email.txt" waarin je je e-mail address plaatst. Geef deze commit de commit message "Add email file"
8. Merge nu alle wijzigingen van de number branch in de main branch.
9. Maak een lege repo aan op github of een gelijkaardige service en koppel deze als remote repo aan je lokale repo. Push de volledige inhoud van de main branch naar je remote repo. (kies een private repo aangezien je naam, email en studentennummer in de repo staan)

Extra oefenmogelijkheid

https://ohmygit.org/

Spelletjesgewijs aan de slag met git.

https://youtu.be/1ffBJ4sVUb4?si=RCNk76i4BHZFYPVT

Nog een goede git uitleg