



**GIT**

Continuous Integration Basics

# Wat is GIT?

---

- **Gedistribueerd** versiebeheersysteem
- Oorspronkelijke gemaakt door **Linus Torvalds** voor de ontwikkeling van de Linux kernel

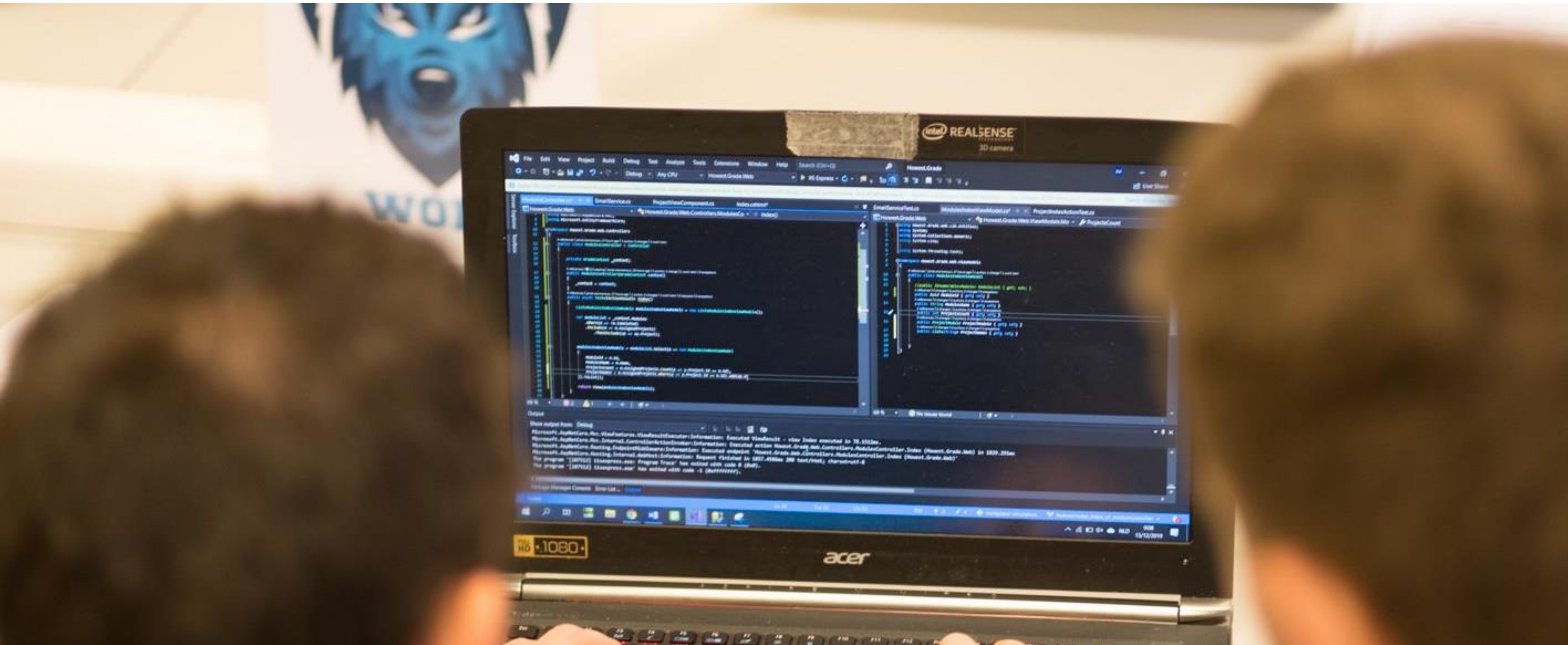


# Waarom versiebeheersystemen?

---

- Samenwerken wordt gemakkelijker
- Opvolgen van veranderingen
- Wie heeft wat gedaan?
- Terugkeren naar vorige versies
- Backup
- Zowel lokale als online repositories mogelijk

# Werken in teams (Workplace Simulation 1920-JH1)





# Werken in teams (Workplace Simulation 1920-JH1)



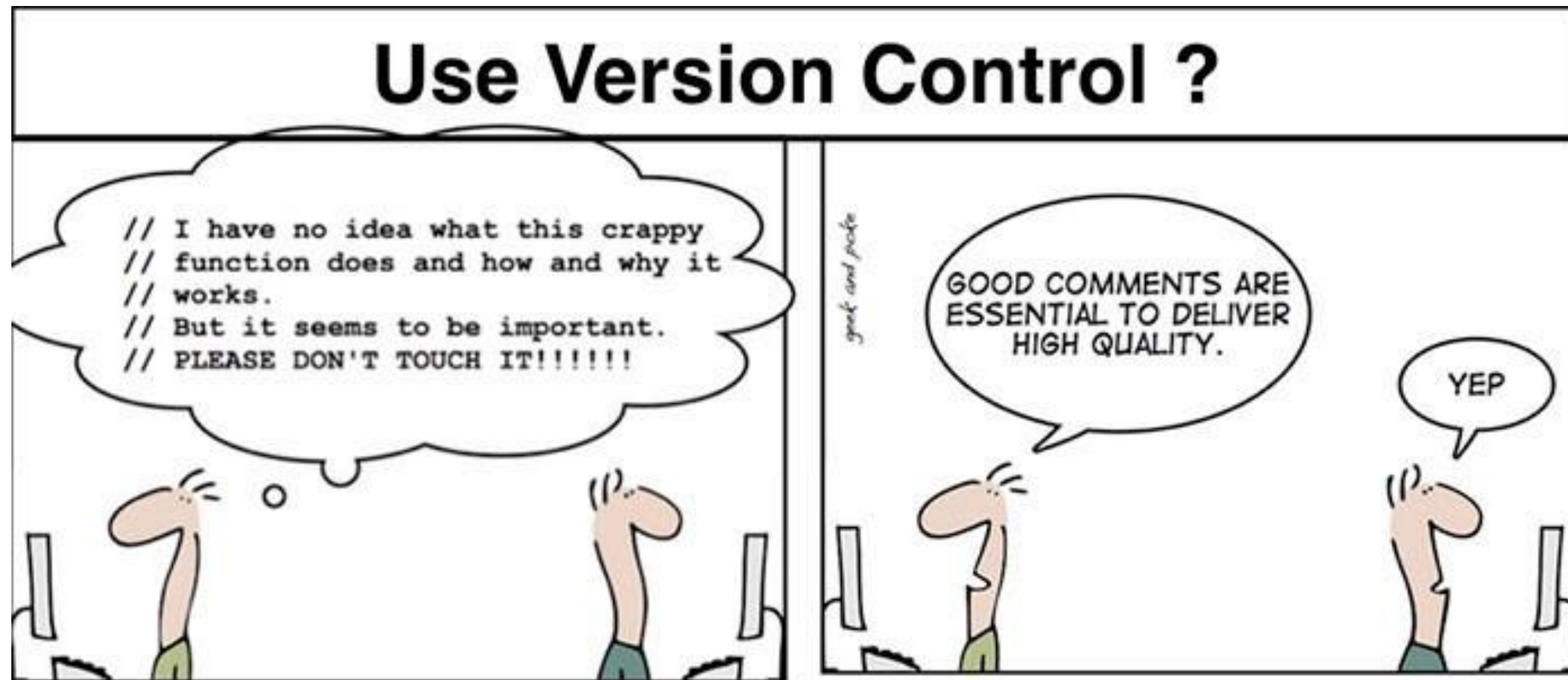


# Werken in teams (Workplace Simulation 1920-JH1)



# Concept versiebeheersystemen

- Opvolging van geschiedenis van de code
- Neemt “snapshots” van de code
- Je beslist zelf wanneer je een “snapshot” maakt door een **commit** te doen
- Bestanden moeten **gestaged** worden voor een **commit**



# Gedistribueerd vs centrale versiebeheersystemen

---

- **Centrale** versiebeheersystemen
  - Aanpassingen worden standaard geüpload naar een server wanneer een nieuwe versie ingediend wordt
  - De vorige versie wordt overschreven
  - Altijd verbinding met de server nodig
- **Voorbeelden** centrale versiebeheersystemen:
  - Subversion (SVN)
  - Team Foundation Service (TFS)





# Gedistribueerd vs centrale versiebeheersystemen

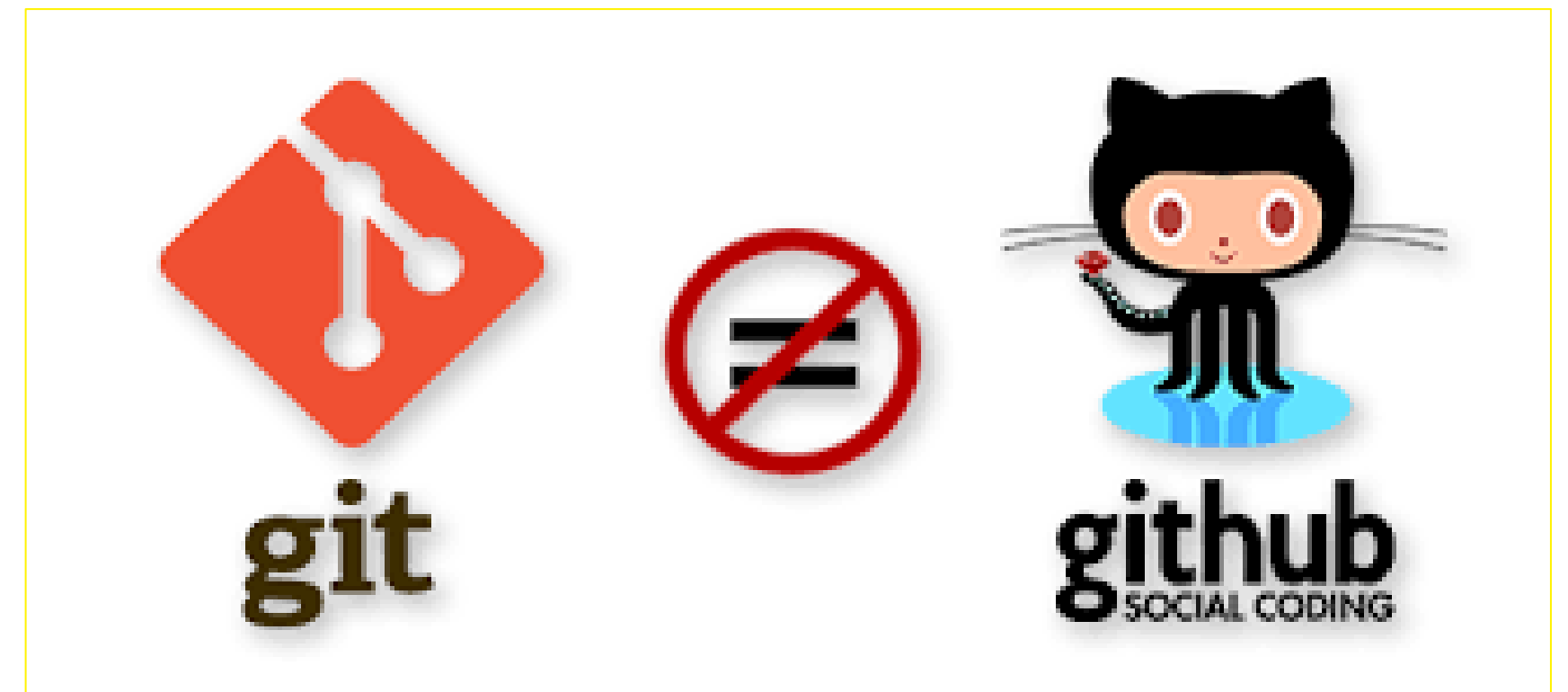
---

- **Gedistribueerd** versiebeheersysteem
  - Aanpassingen worden ingediend in een lokale kopie
  - Op elk gewenst moment kan deze lokale kopie gesynchroniseerd worden met de server
  - Aanpassingen kunnen lokaal gebeuren, dus geen constante verbinding met de server nodig
  - Alle versies blijven bewaard
- **Voorbeelden** gedistribueerd versiebeheersysteem:
  - Git
  - Mercurial



# Git & GitHub

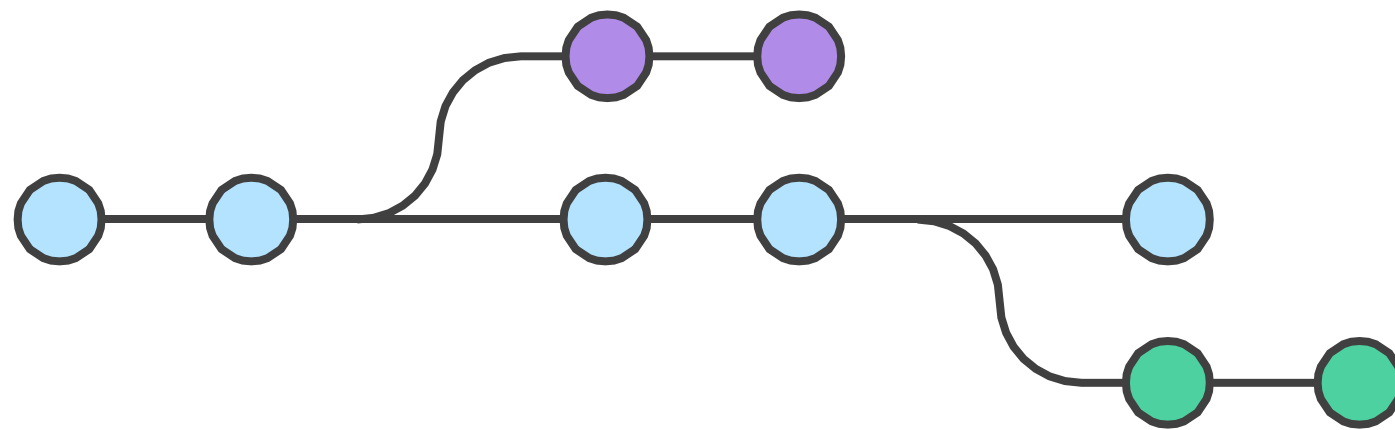
- **Git**
  - Is een versiebeheersysteem
- **GitHub**
  - Is een webapp rond git functionaliteit
  - Online repositories (= remote)
- **Alternatieven** GitHub
  - Bitbucket
  - GitLab
  - DevHub
  - GitKraken
  - ...



# Enkele termen

---

- **Repository**  
Plaats waar je code staat
- **Commit**  
Toevoeging van code (aan een branch) = snapshot
- **Branch**  
Onafhankelijke aftakking van de code base (tijdelijk of om later te “mergen”)

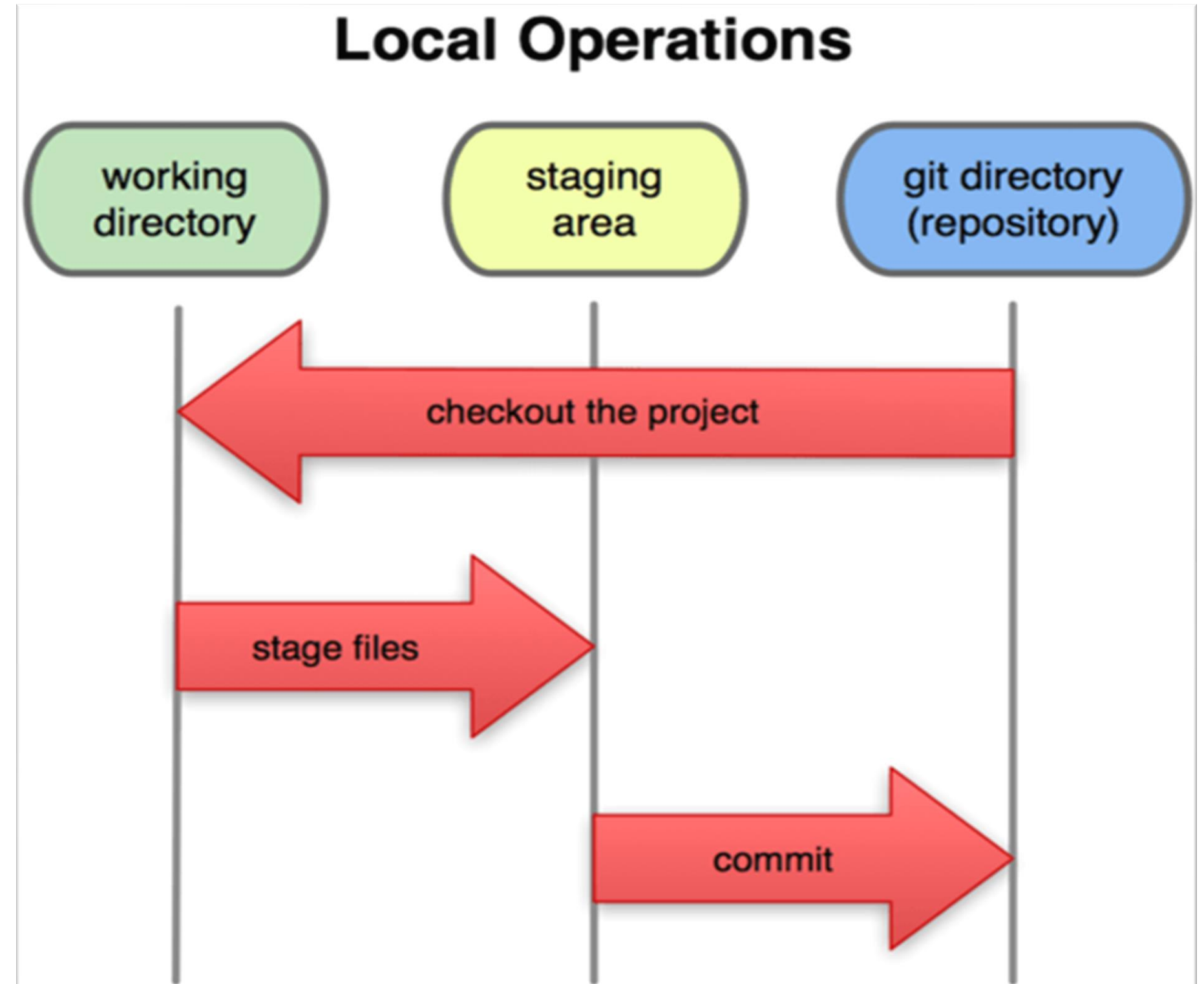




# Lokale repositories

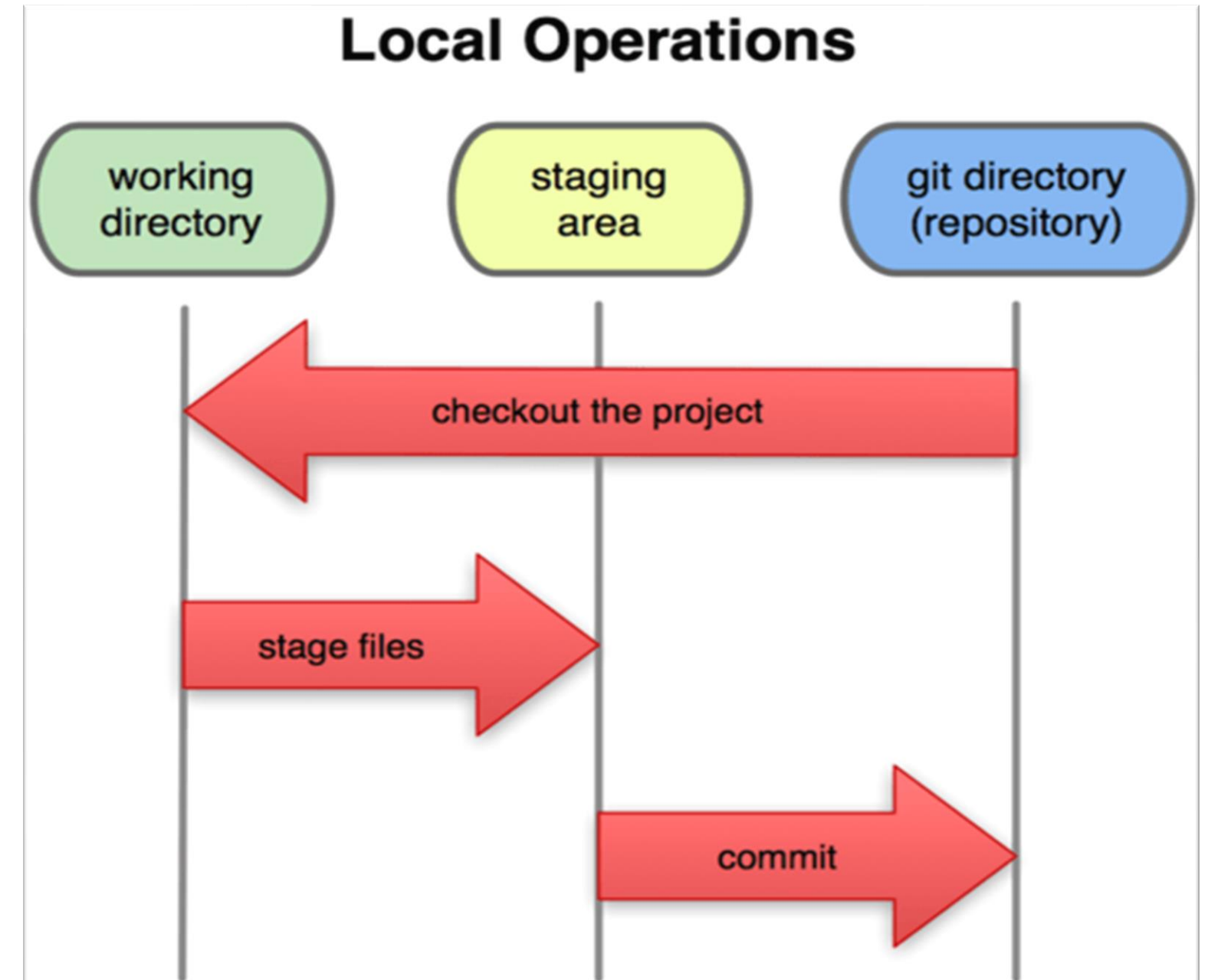
# De drie hoofdonderdelen

- Git directory (repository)
- Staging Area
- Working directory



# Algemene workflow met Git

- Je bewerkt bestanden in je working directory.
- Je bereidt de bestanden voor (staged), waardoor momentopnames (snapshots) worden toegevoegd aan de staging area.
- Je maakt een commit.  
Een commit neemt alle snapshots van de staging area en bewaart die voorgoed in je Git directory.

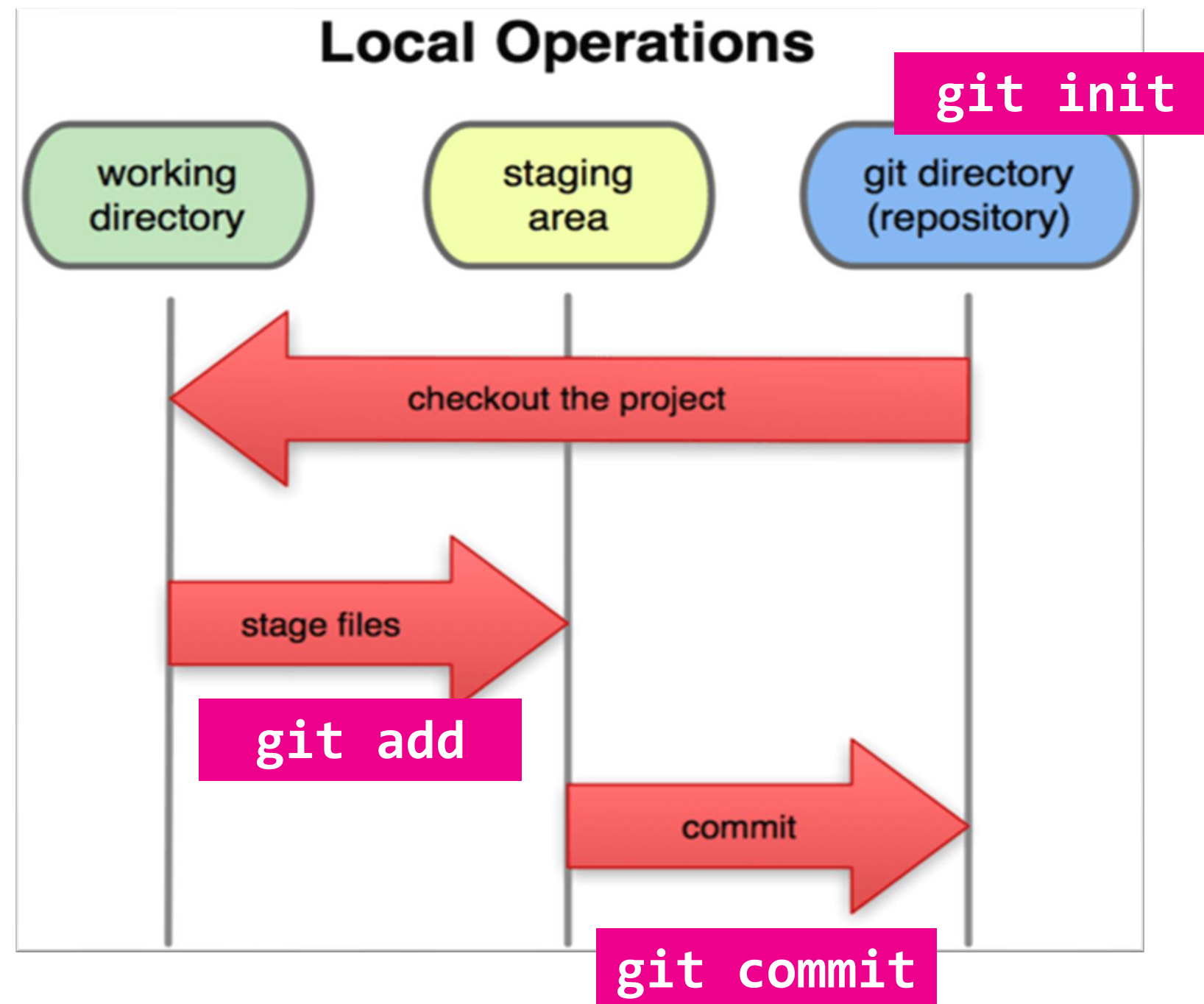




# Enkele commando's lokale repository

Commando	Resultaat
<code>git init</code>	Activeer git versiebeheer in de huidige map (= nieuwe lokale repo starten).
<code>git add &lt;file&gt;</code>	Voeg de wijzigingen in het bestand <file> toe aan de staging area.
<code>git add &lt;folder&gt;</code>	Stage de wijzigingen in alle files in de map <folder>.
<code>git commit -m &lt;message&gt;</code>	<p>Maak een nieuwe commit (snapshot) die alle wijzigingen bevat die op dit moment in de staging area klaarstaan.</p> <p>Met de parameter -m wordt een gepaste commit message meegegeven, bv. <code>git commit -m "Fix login issues"</code></p>
<code>git status</code>	Geeft een overzicht van de bestanden met wijzigingen, inhoud staging area, etc. Zeer nuttig om regelmatig uit te voeren zodat je geen fouten maakt!

# Enkele commando's lokale repository

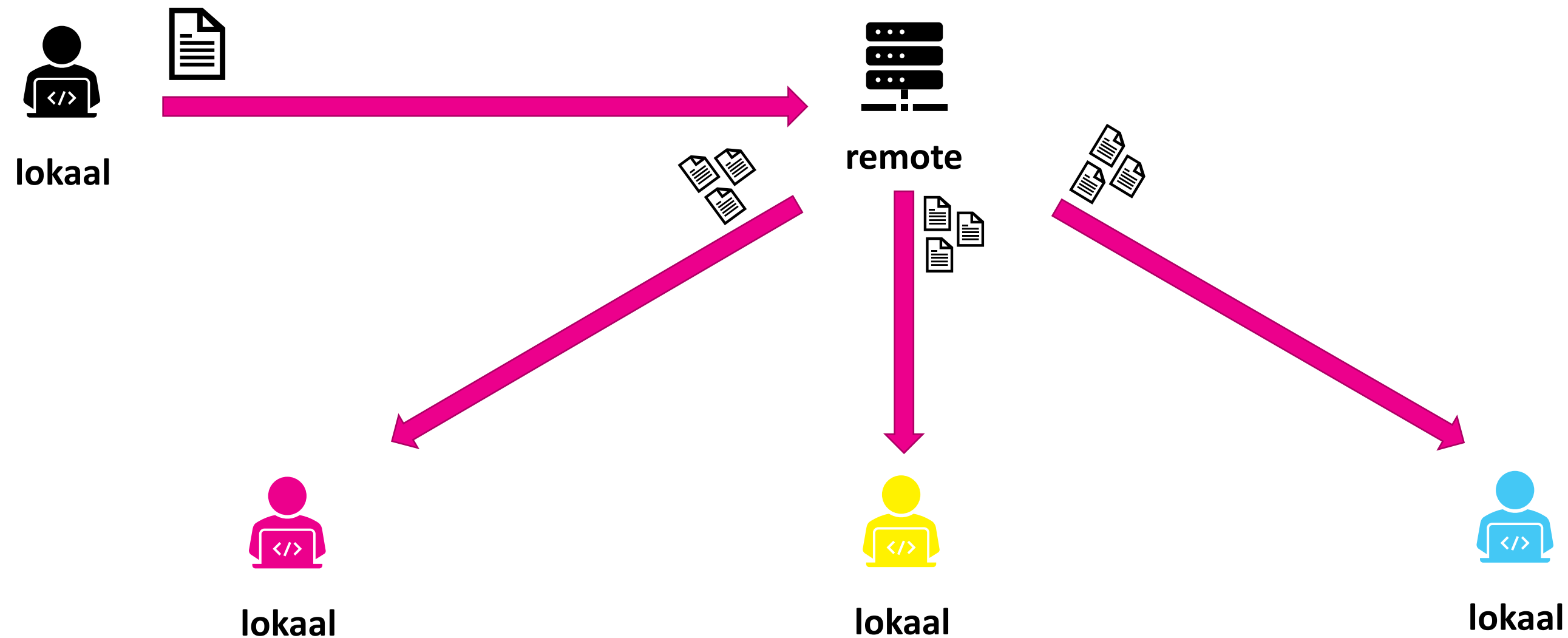


# Remote repositories



# Remote repository

Een **online** git repository (ergens op een server, bv. GitHub)



# Local vs remote git repository

---

## **Local** git repository

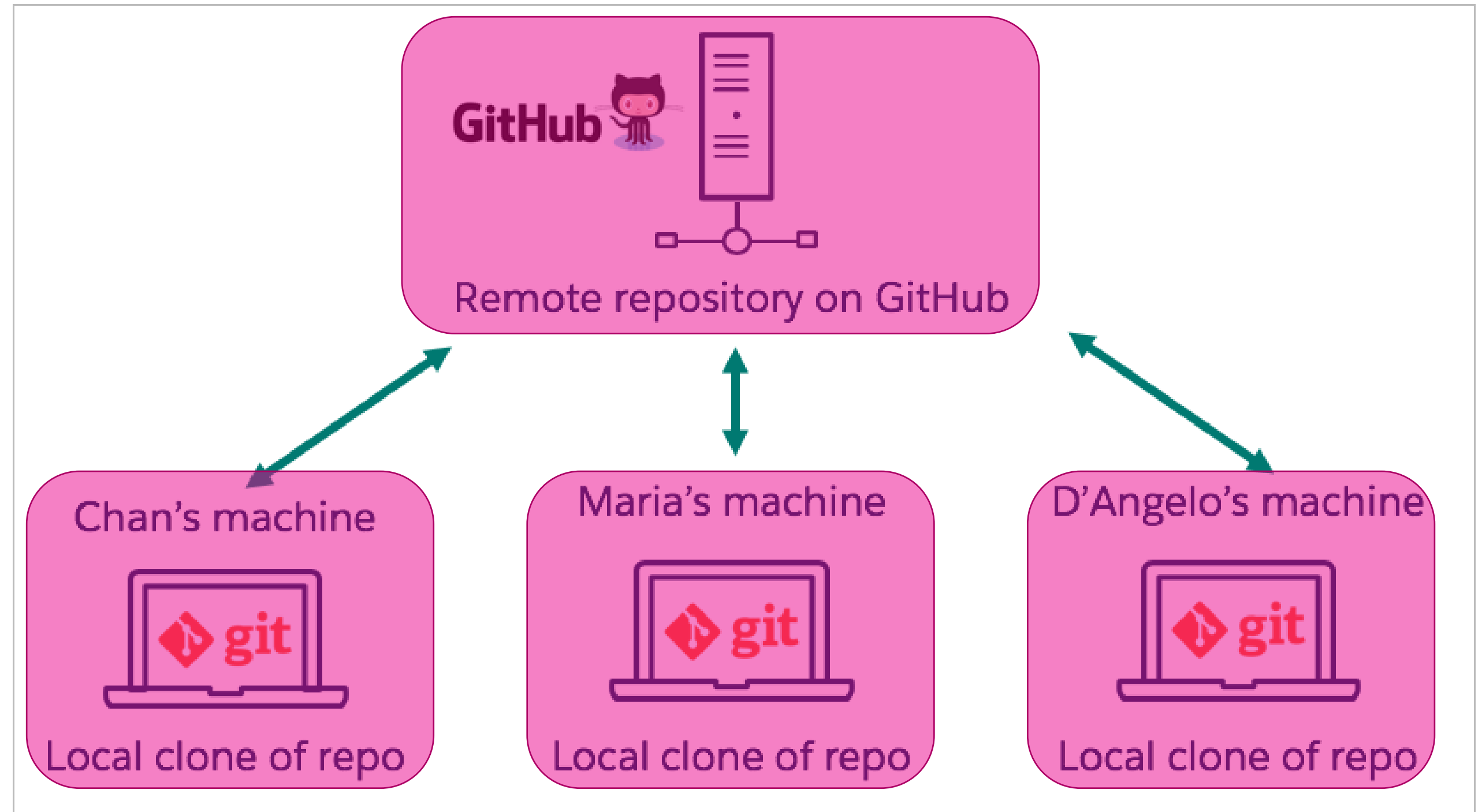
- Bevindt zich lokaal op onze computer

## **Remote** git repository

- Bevindt zich online op een server
- Bv. GitHub, Bitbucket, ...

# Local vs remote git repository

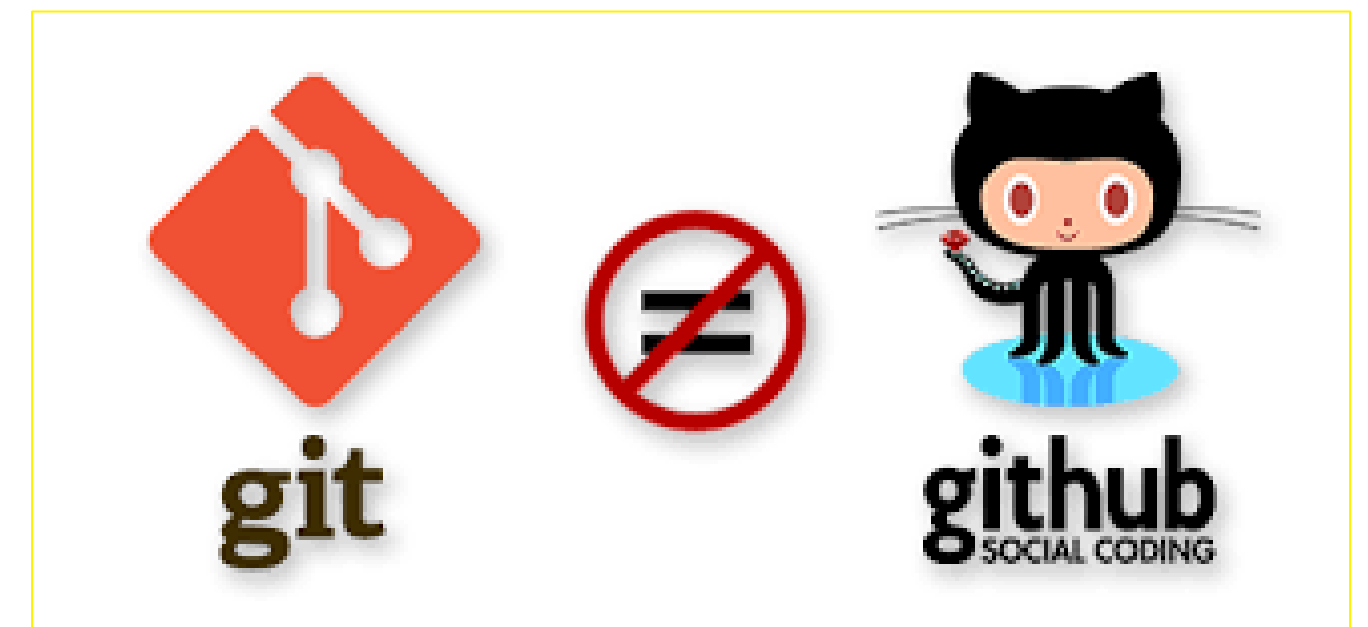
- Remote
- Local





# GitHub

- Waarom GitHub?
  - Populair
  - Student develop pack voor studenten
  - Gratis voor studenten (aantal private repositories)
- Website gebouwd rond het git versiebeheersysteem
- De locatie waar we onze code zullen opslaan om te kunnen werken aan deze code
  - Op meerdere plaatsen
  - Met meerdere developers
- GitHub != git



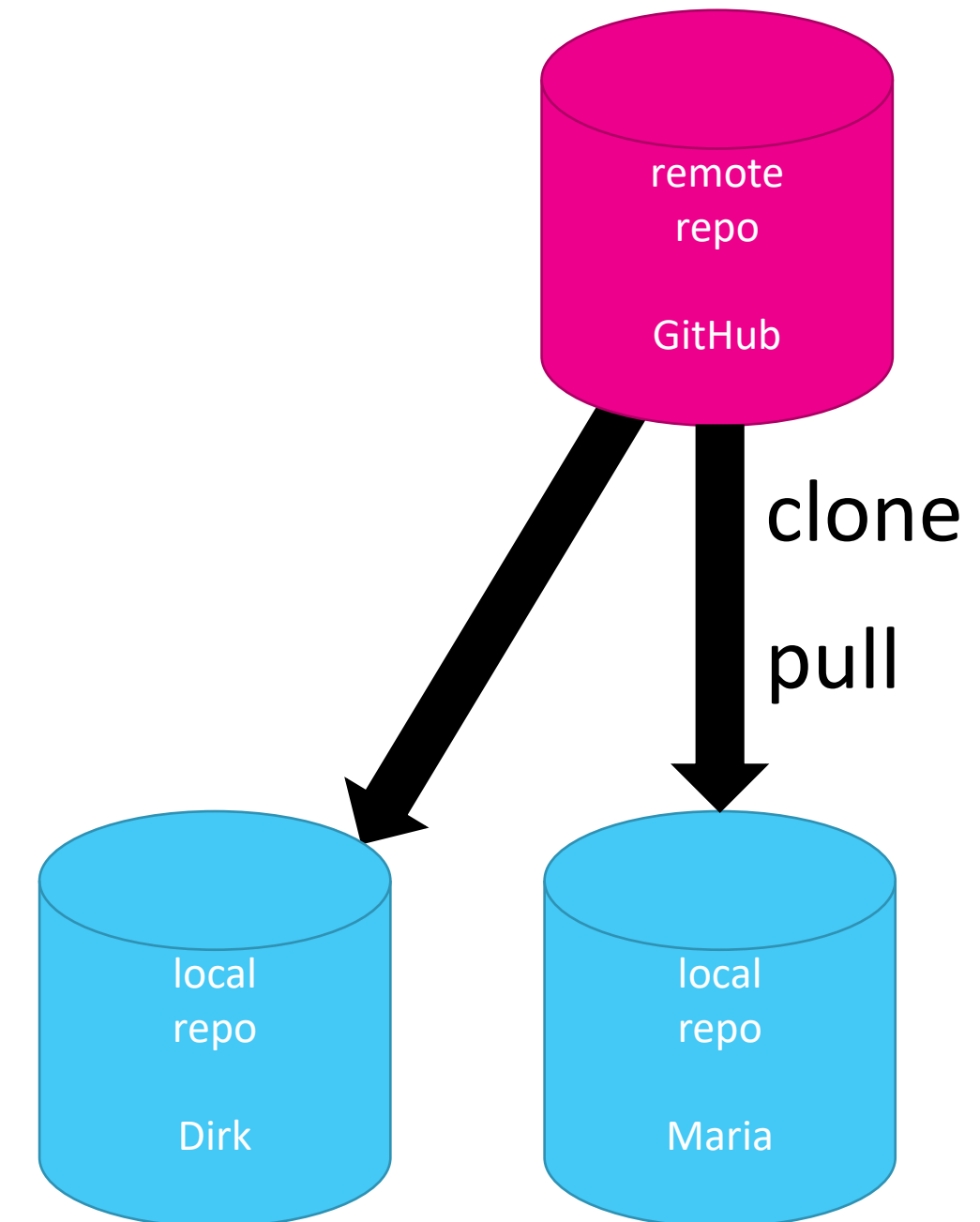
# Alternatieven

---

- Bitbucket
- GitLab
- DevHub
- Beanstalk
- SourceForge
- Apache Allura
- Cloud Source (Google)
- WS CodeCommit
- GitKraken
- ...

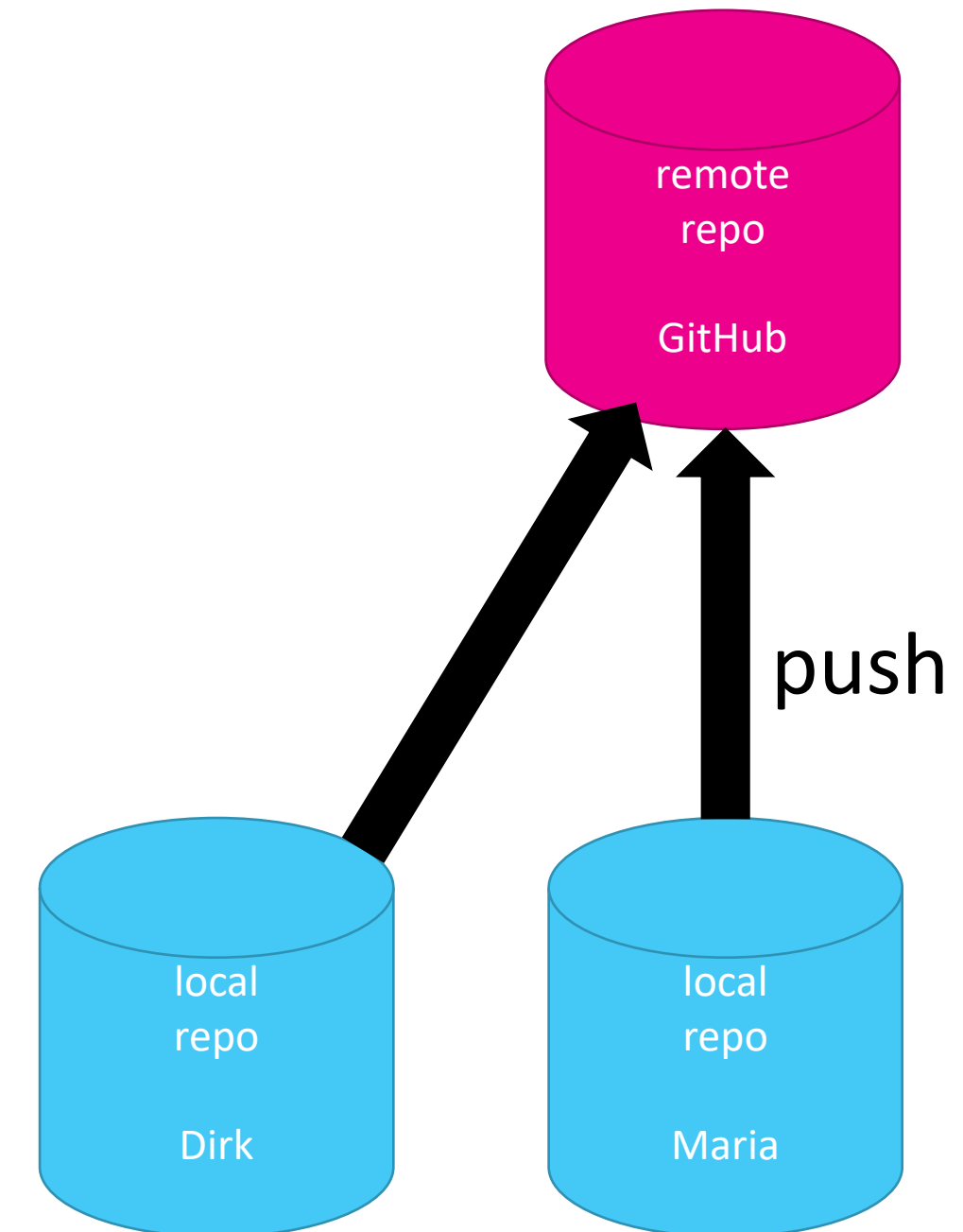
# Van Remote naar Local repository

- Nog geen local repo aanwezig?
  - Clone (volledige remote downloaden)
  - Local is nu gelijk aan remote
- Wel reeds een local repo aanwezig?
  - Pull (wijzigingen remote downloaden)
  - Local is nu gelijk aan remote
- Meerdere gebruikers (local repos) met één remote
  - Clone
  - Pull



# Van Local naar Remote repository

- Lokale wijzigingen naar de remote sturen
  - Push
  - Remote is nu gelijk aan local
- Meerdere gebruikers
  - Push
  - Remote is gelijk aan local met laatste push
  - Merge conflicts?





# Termen

---

## Clone

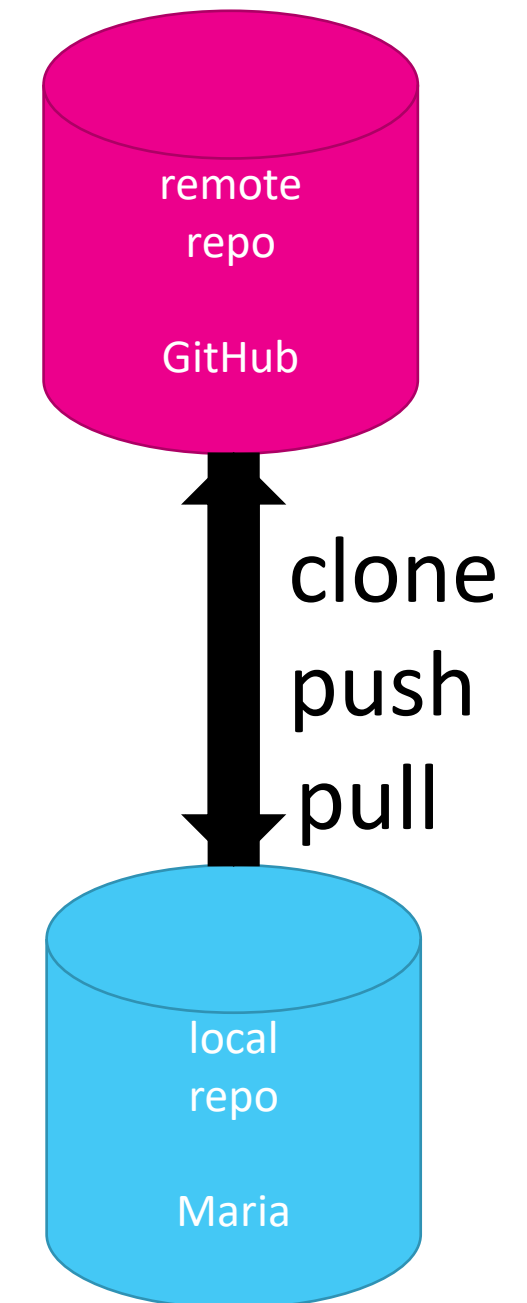
- Remote clonen naar local

## Push

- Van local naar remote sturen

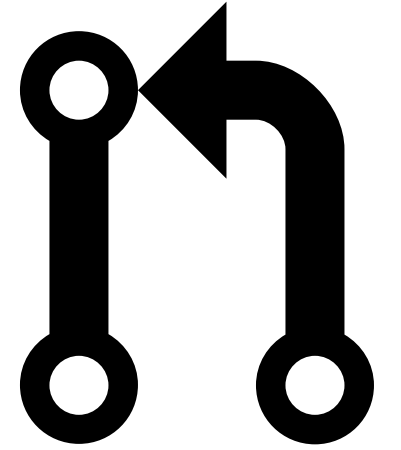
## Pull

- Van remote naar local

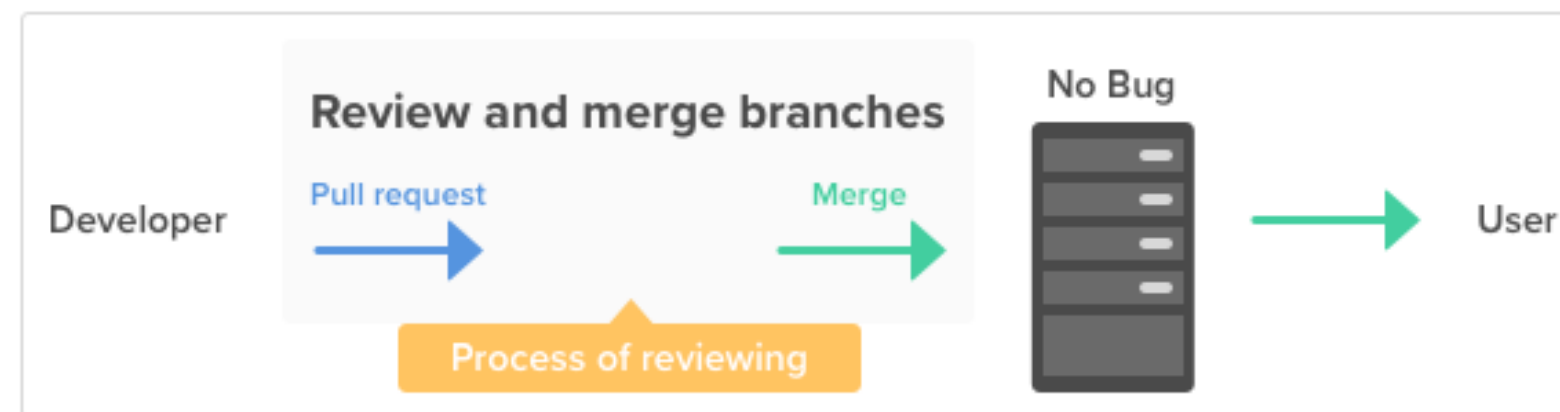


# Samenwerken aan één repository

- Maak gebruik van **Pull Requests**
- De eigenaar/verantwoordelijke beslist of je wijzigingen al dan niet opgenomen worden in de remote
  - Approve
  - Decline
  - Request Changes
- Werk in **branches!**
  - Komt later aan bod



Development without pull request



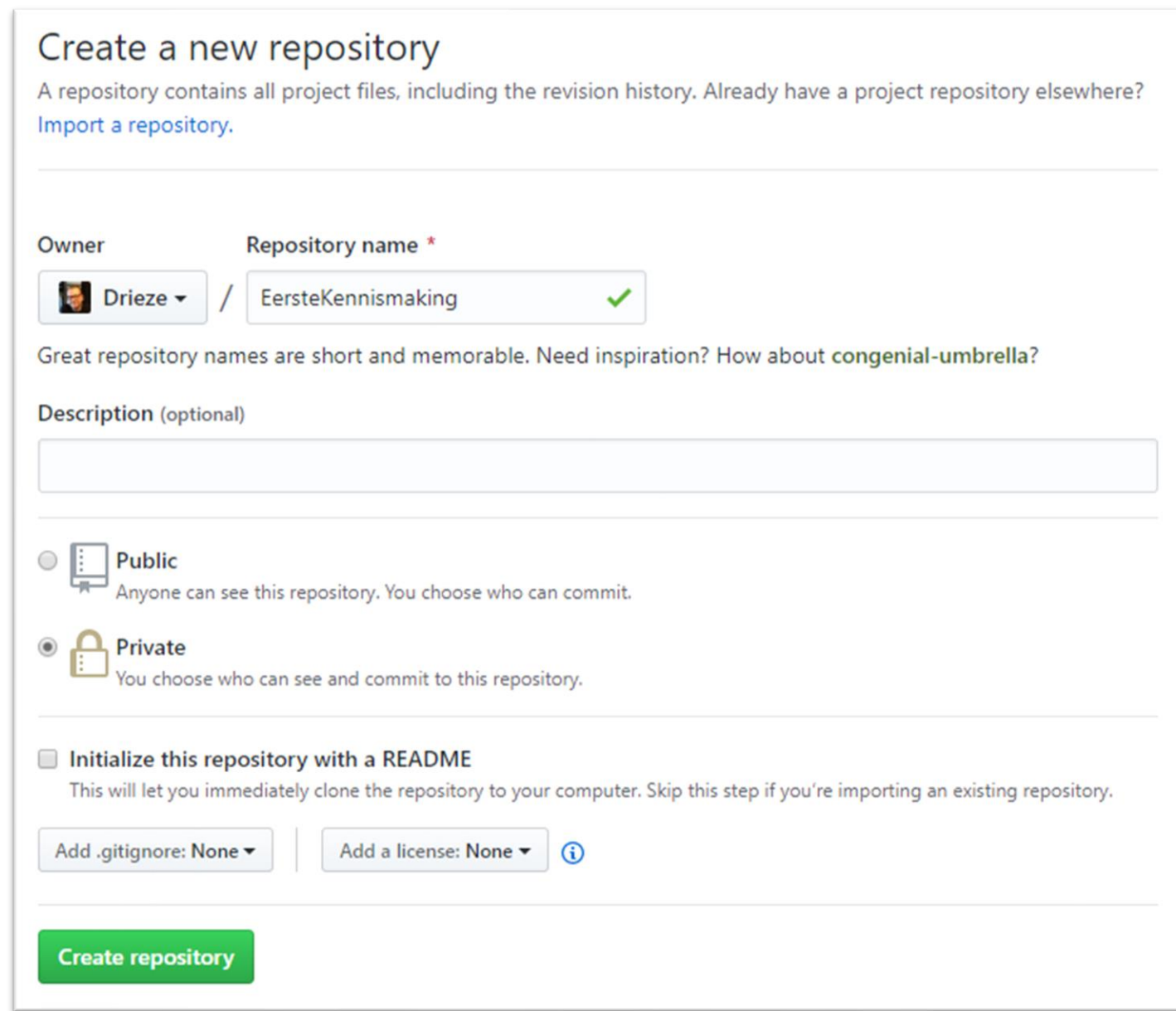
Development with pull request

# Enkele commando's

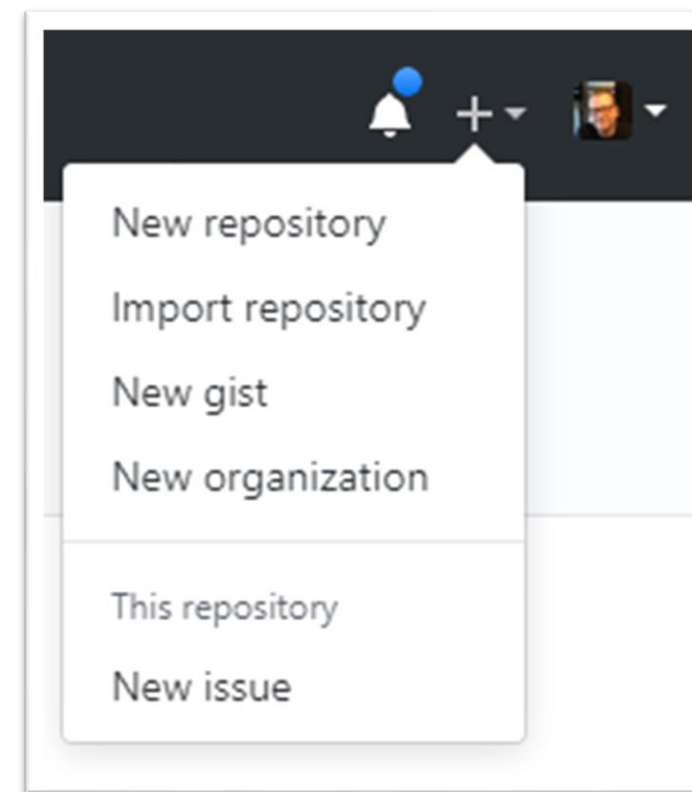
Commando	Resultaat
<code>git clone &lt;url&gt;</code>	Maak een lokale kopie (clone) van de remote repo op de gegeven locatie
<code>git remote</code>	Toon remote repo's gelinkt aan je lokale repo
<code>git remote add origin &lt;url&gt;</code>	Voeg remote toe met alias <i>origin</i> en gegeven <i>URL</i>
<code>git remote rm origin</code>	Verwijder de remote met alias <i>origin</i> uit de lokale repo
<code>git push</code>	Wijzigingen in lokale repo (nieuwe commits) doorsturen naar remote repo
<code>git push -u origin master</code>	Bij eerste keer pushen is het nodig om de remote (hier <i>origin</i> ) en branch (hier <i>master</i> ) mee te geven waar je naar wil pushen—via de optie <code>-u</code> (of voluit <code>--set-upstream</code> ) zorg je ervoor dat dit onthouden wordt voor volgende pushes
<code>git pull</code>	Nieuwe wijzigingen (commits) van remote repo binnentrekken in lokale repo

# Voorbeeld 1: lokale repo manueel linken aan nieuwe remote

## 1. Maak lege **remote** repo op GitHub

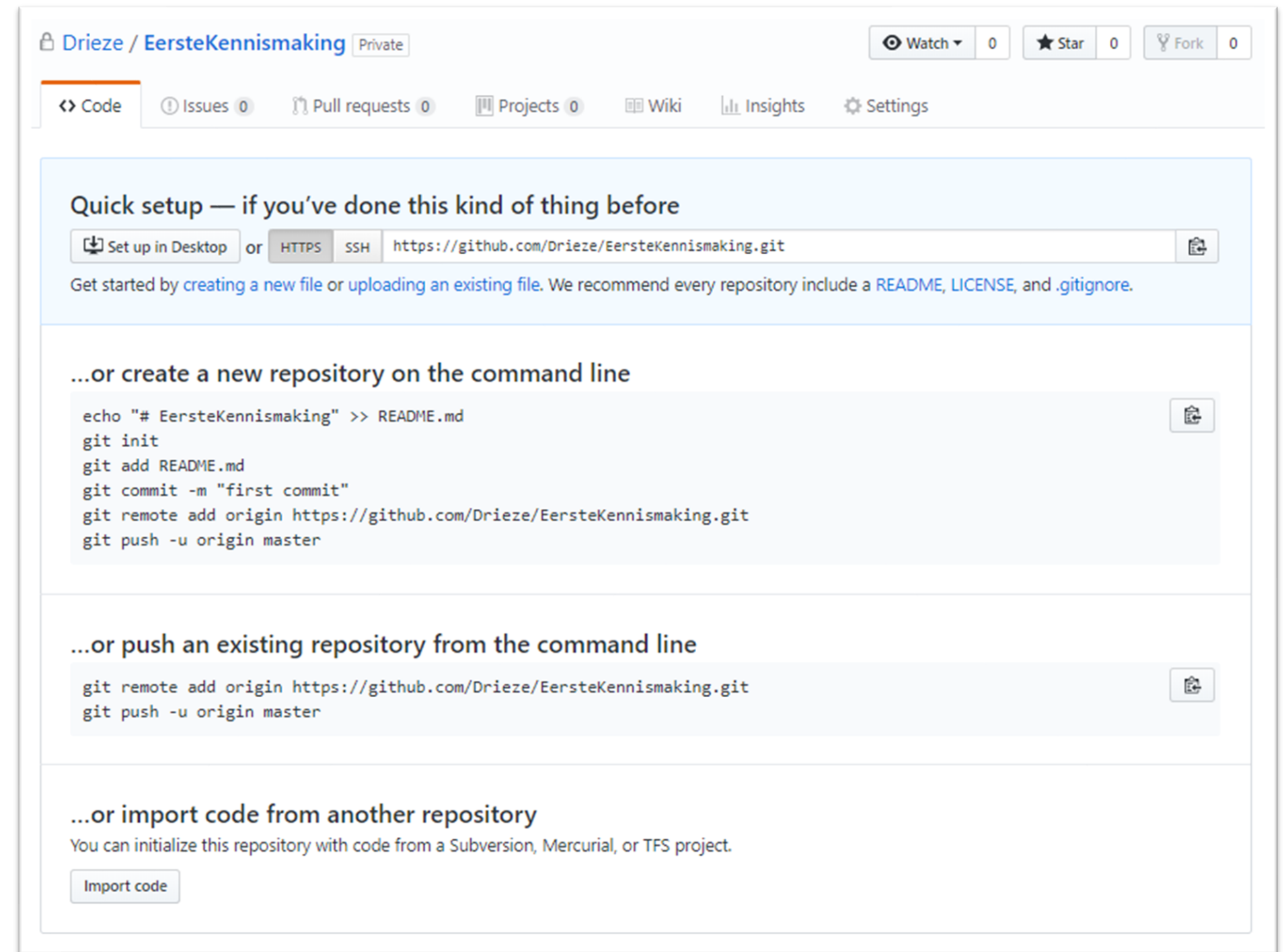


The screenshot shows the 'Create a new repository' form on GitHub. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is. Below this, there are fields for 'Owner' (set to 'Drieze') and 'Repository name' (set to 'EersteKennismaking' with a green checkmark). A hint suggests repository names should be short and memorable. There is an optional 'Description' text area. Under 'Visibility', the 'Private' option is selected. At the bottom, there is a checkbox for 'Initialize this repository with a README' and two dropdown menus for '.gitignore' and 'Add a license'. A green 'Create repository' button is at the bottom.



# Voorbeeld 1: lokale repo manueel linken aan nieuwe remote

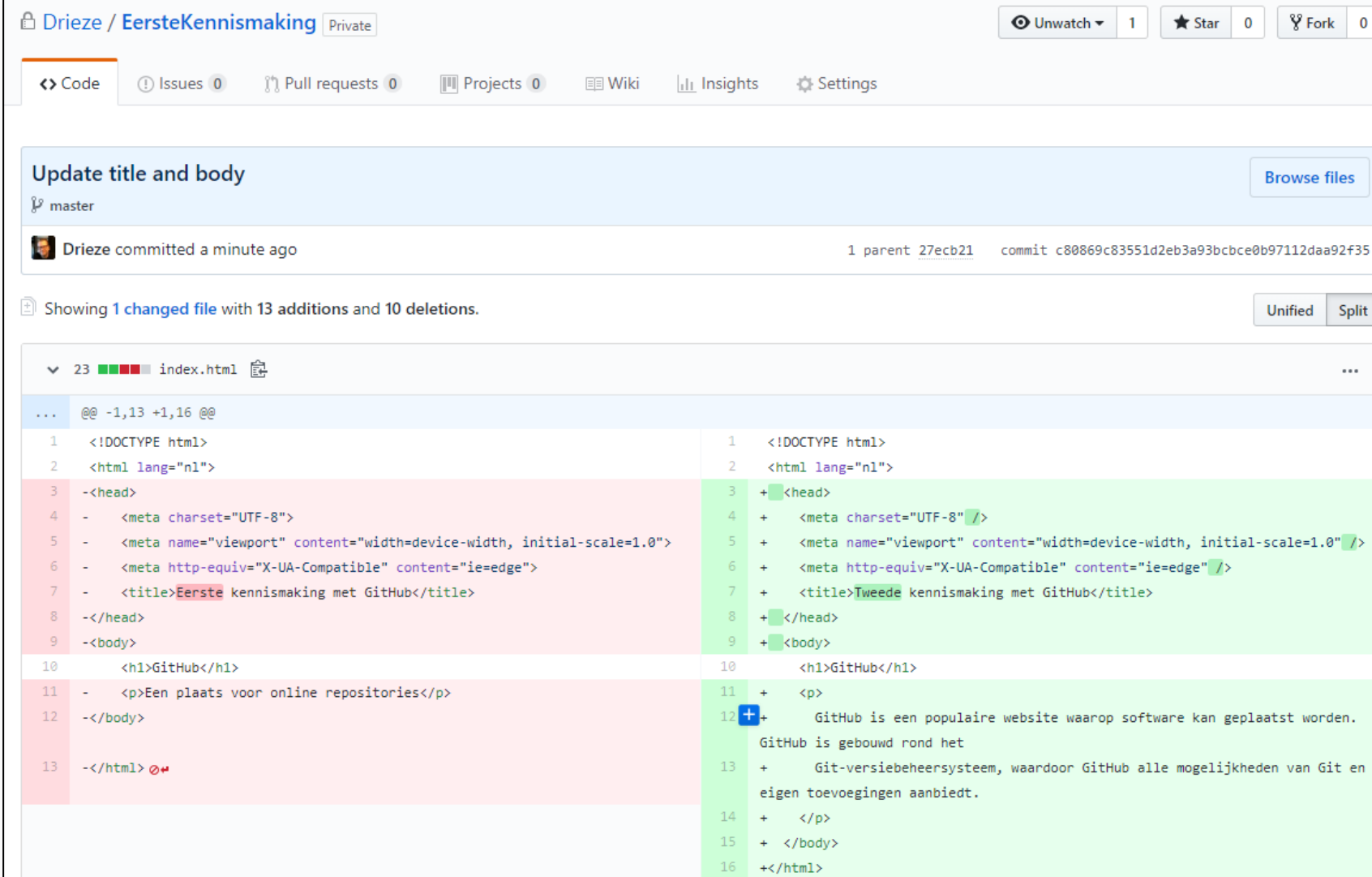
2. Maak nieuwe **lokale** repo
3. Voeg een paar **bestanden** toe
4. Voeg **remote** repo toe (origin)
  - Kopieer URL van GitHub
5. **Controleer** of remote goed werd toegevoegd
6. **Push** lokale commits naar remote
7. Check resultaat op **GitHub**





# Voorbeeld 2: clone van bestaande remote repo

1. Verwijder je lokale repo—**NIET** de remote repo
2. Voer een clone uit van de remote repo
3. Controleer lokale repo: alle bestanden en geschiedenis nog aanwezig
4. Voer wijzigingen door aan bestanden
5. Commit & push naar remote
6. Check resultaten op GitHub



The screenshot shows a GitHub repository page for 'Drieze / EersteKennismaking'. The commit history shows a commit by 'Drieze' a minute ago. The diff view shows changes to 'index.html'. The left side shows the original code with line numbers 1 to 13. The right side shows the modified code with line numbers 1 to 16. The changes include adding a new paragraph about GitHub and updating the title from 'Eerste' to 'Tweede'.

```
@@ -1,13 +1,16 @@
1 <!DOCTYPE html>
2 <html lang="nl">
3 <head>
4 - <meta charset="UTF-8">
5 - <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 - <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 - <title>Eerste kennismaking met GitHub</title>
8 </head>
9 <body>
10 <h1>GitHub</h1>
11 - <p>Een plaats voor online repositories</p>
12 </body>
13 </html>
14 + <p>
15 +   GitHub is een populaire website waarop software kan geplaatst worden.
16 +   GitHub is gebouwd rond het
17 +   Git-versiebeheersysteem, waardoor GitHub alle mogelijkheden van Git en
18 +   eigen toevoegingen aanbiedt.
19 + </p>
20 </body>
21 </html>
```

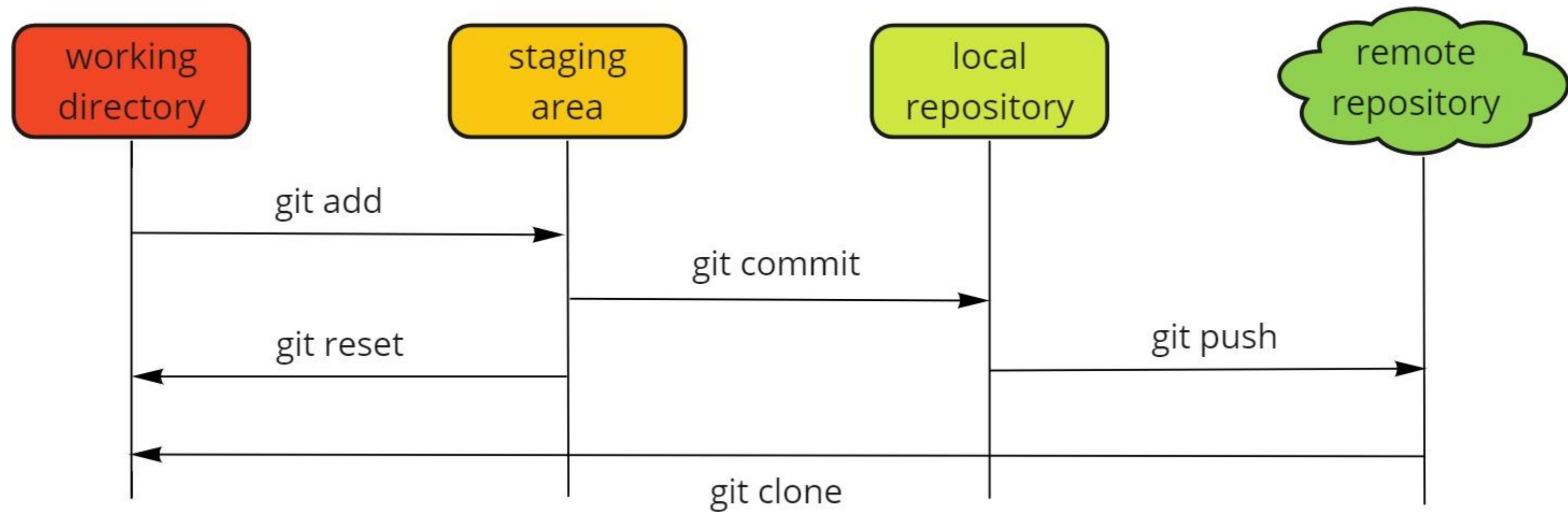
# Voorbeeld 3: meerdere developers (contributors)

---

1. **Kandidaat proefpersoon** nodig → collaborator toevoegen
2. **Developer 2:**
  - Clone remote repo
  - Voer enkele wijzigingen door in lokale repo
  - Synchroniseer commits naar remote repo
  - Check resultaat op GitHub
3. **Developer 1:**
  - Voer wijzigingen door in lokale repo
  - Probeer te pushen?! → eerst pull nodig van laatste remote changes!
  - Pull & push
  - Check eindresultaat op GitHub
4. **Developer 2:**
  - Pull wijzigingen van developer 1

**CONTINUOUS  
integration!!**

## Grafisch overzicht commando's



miro