# Stappenplan kerntaak 4

**Let op!** Het maken van een git repository kan worden overgeslagen op het examen want dit is al voor je gedaan.

### Stap 1:

Maak een repository aan op git door de volgende stappen te volgen:

1. Klik op de groene knop *New Project* of ga via de navigatie naar *Projects -> Your Projects -> New Project* (groene knop rechtsboven).
2. Je komt op de pagina voor het aanmaken van een nieuw project. Bovenin zie je 4 tabbladen. Meestal zal je de eerste gebruiken (Blank project) maar soms zal je kiezen voor de derde, *Import project*.
3. Voor nu gebruiken we de eerste, *Blank project*.
   1. Geef je project een naam, bijvoorbeeld *Portfolio* of *WebShop*.
   2. Een *description* is optioneel. Hier kan je bijvoorbeeld omschrijven wat er binnen dit project te zien is.
   3. *Visibility level*: dit wilt zeggen; wie kan het project zien? Alle docenten van de opleiding zijn admin, zij kunnen altijd jouw project zien, ook al staat die op private.
      1. **Private**: alleen jij, de docenten en eventueel toegevoegde leden kunnen je project zien.
      2. **Internal**: iedereen met een account op de GitLab server kan je project zien.
      3. **Public**: iedereen met de link naar jouw project kan je project zien. In principe kan iedereen het dus zien. Dit is handig als je bijvoorbeeld gaat solliciteren en je wilt je werk laten zien aan je stagebedrijf.
4. Klik **niet** op het vinkje met *Initialize repository with a README*.
5. Klik op de groene knop met *Create Project*.
6. Profit.

### Stap 2:

Maak de tag aan met de titel 1.0 dat is namelijk de eerste release versie van dit project en vanaf hieruit zal worden verder gewerkt. Door de tag te creëren maak je het de developer makkelijk om via de branch weer terug te gaan naar een oudere versie van het project.

### Stap 3:

Maak van alle documenten en het project zelf (De code, database, etc..) een versie 1.0 als dit nog niet gedaan is.

Stap 4:  
Clone de git repository naar je werkomgeving.

### Stap 5:

Zet alle bestanden die bij het project horen in de (geclonede) git repository op je werkomgeving.

### Stap 6:

Voeg de bestanden toe aan je git repository dit kan op de volgende manieren:

* Commando: git add <bestandsnaam>
  + Een bestand klaarzetten voor cmmit
* Commando: git add .
  + Alle bestanden klaarzetten voor commit
* Commando: git status
  + Laat de huidige status van de bestanden zien
    - Als het toevoegen goed is gegaan is het bestand groen

### Stap 7:

Commit je bestand(en) naar git op de volgende manier:

* Commando: git commit -m
* Commit: Een bestand committen
* -m sstaat voor “message”. Als je een commit wilt doen, moet er **ALTIJD** een message staan. De message’s moeten ook serieus en duidelijk.
* Message kan zijn: Navigatie voor website afgemaakt.
* Message kan zijn: Project klaar voor oplevering.
* Message kan zijn: Laatste bugs gefixt voor klant.

### Stap 8:

Nu ben je klaar om de bestanden naar git te pushen doe dit op de volgende manier:

* Commando: git push -u origin master
  + Alle bestanden worden naar Gitlab ‘Gepusht’
* Git Bash zal nu vragen of jouw SSH sleutel mag worden verbonden met gitlab.rijnijssel
* Type: yes
* Je krijgt een WARNING te zien dat gitlab.rijnijssel is toegevoegd aan betrouwbare sites

### Stap 9:

Ververs je browser en kijk of alle bestanden aan git zijn toegevoegd.

### Stap 10:

Maak op gitlab onder het kopje issues een board aan een board is een overzicht van al je issues. Een board moet de volgende kolommen bevatten:

* Open: je *backlog*. Alles wat nog open staat om opgepakt te worden.
* To Do: issues waarvan jij zegt; deze ga ik oppakken.
* Doing: issues waar jij mee bezig bent.
* Review: issues zijn klaar om besproken te worden met ontwikkelteam, docent, product owner (/opdrachtgever), stagebegeleider etc.
* Closed: issues zijn opgelost en goedgekeurd.

**Let op!** De kolommen Open & Closed staan standaard al in een board toegevoegd je moet zelf de overige kolommen nog toevoegen dit doe je door labels aan te maken zie hier voor **Stap 12.**

### Stap 11:

Plan een gesprek in met je leidinggevende en neem meerdere (minimaal vier) tickets in ontvangst.

### Stap 12:

Maak labels aan voor de issues doe dit op de volgende manier:

* Ga naar je repository op gitlab.rijnijssel.nl
* Links in het menu: Issues > Labels
* Aanmaken nieuwe label:
  + Klik rechtsboven op de groene knop New label.
  + Voer *Title* en *Background* color in.
  + *Description* is optioneel, hier zou je een omschrijving kunnen plaatsen over een label.
  + Klik links onderin op *Create label.*

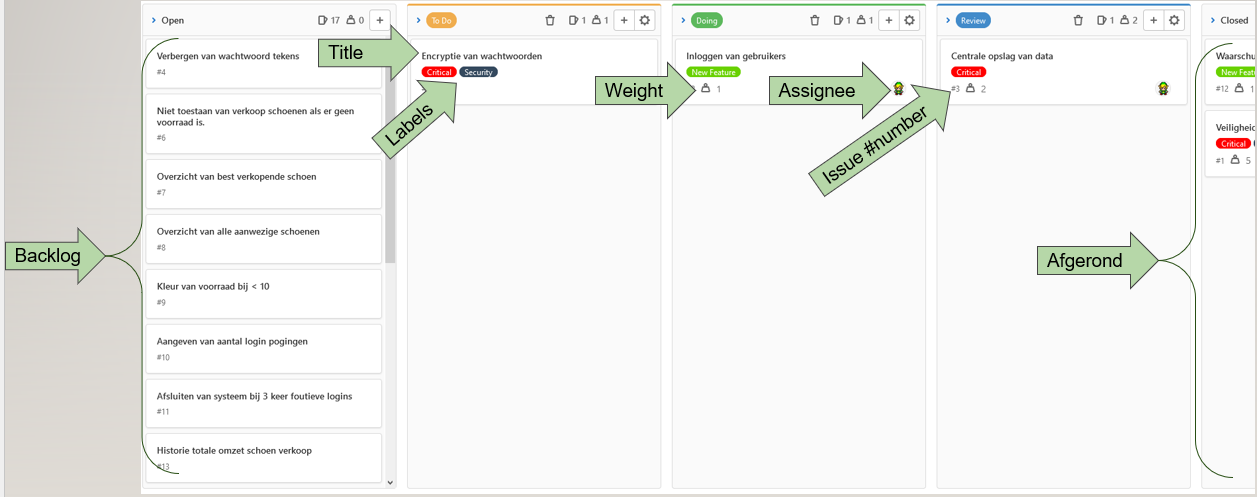
Gebruik voor de titels van de labels de titels die in de opdracht omschrijving staan of gebruik je eigen sjabloon voorbeelden van titels zijn: New content, New function, Bugs, Critical Error etc..

### Stap 13:

Voeg alle issues die je hebt gekregen van je leidinggevende toe aan je repository op gitlab.rijnijssel.nl op de volgende manier:

* Klik links in het menu op Issues
  + Klik rechtsboven op *New issue.*
  + Voer *Title, Assignee, Labels* en *Weight* in.
    - Title: titel van je Issue, bijvoorbeeld: navigatiemenu ontwikkelen.
    - Assignee: wie gaat deze issue oplossen?
    - Labels: wat voor een type issue is dit?
    - Weight: hoeveel tijd is hiervoor nodig (schatting)? Reken in uren, laagste is dus 1 uur. Geen kommagetallen!
  + Overige gegevens zijn optioneel
  + Klik op *Submit issue.*

Je board ziet er nu als volgt uit:



### Stap 14:

Bedenk oplossingen voor alle tickets gebruik hierbij de uitgewerkte applicatie en database. Houd rekening met de randvoorwaarden zoals vermeld in het programma van eisen, het functioneel ontwerp en het technisch ontwerp. Zorg er voor dat de oplossingen duidelijk beschreven zijn op de plaats waar de tickets opgeslagen zijn. Dit doe je op de volgende manier:

* Navigeer naar je board op gitlab.rijnijssel.nl
* Klik op de issue en scroll naar beneden
* Hier is een tekst vak te zien omschrijf hier je oplossing
* Klik op “Comment”

### Stap 15:

Overleg met je leidinggevende welke van de tickets je gaat realiseren. (Dit zijn er minimaal twee.) En zet deze in het kolom “To Do”

### Stap 16:

Realiseer de gekozen oplossingen voor de toegewezen tickets in de applicatie en/of de database. Zet de ticket waarmee je bezig bent in de kolom “Doing”

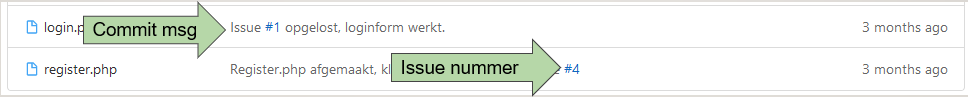
**Let op!** Er kan maar 1 ticket (issue) te gelijk in “Doing” staan.

### Stap 17:

Documenteer de doorgevoerde wijzigingen op de met de leidinggevende overeengekomen wijze. En verander het versie nummer van ieder bestand dat is aangepast

### Stap 18:

Commit de bestanden die je hebt aangepast voor de ticket (issue) naar git kijk voor meer uitleg over hoe je moet committen naar **stap 7.** De volgende stappen moeten nog worden toegevoegd voor het committen van een issue:

* Binnen de quotes ‘’mesage’’ kan je verwijzen naar een issue nummer.
* Door dit te doen zie je in GitLab, in de *commit-tekst*, je issue nummer terugkomen.
* Dit is gelijk een *link*, zodat je naar het betreffende issue navigeert.
* Voorbeeld: git commit -m “Issue nummer #3 is opgelost.”

### Stap 19:

Nu ben je klaar om de bestanden naar git te pushen doe dit op de volgende manier:

* Commando: git push -u origin master
  + Alle bestanden worden naar Gitlab ‘Gepusht’
* Git Bash zal nu vragen of jouw SSH sleutel mag worden verbonden met gitlab.rijnijssel
* Type: yes
* Je krijgt een WARNING te zien dat gitlab.rijnijssel is toegevoegd aan betrouwbare sites

### Stap 20:

Maak wederom een tag aan (doe dit nadat pas nadat je de bestanden naar git hebt gepusht!) tags maak je aan om versies van je software te kunnen onderscheiden. Dit doe je op de volgende manier:

* Ga naar je repository op gitlab.rijnijssel.nl
* Links in het menu: repository > tags
* Klik op de groene knop rechtsboven met *New tag*

Vul de volgende gegevens in:

* Tag name: vul hier een versienummer in.
* Create from: master
* Message: optioneel, kan een beschrijving omvatten waar de tag over gaat
* Release notes: optioneel, dit maakt een public release van je code tot dat moment, je kan hier eventueel notes plaatsen (denk aan Patch Notes)
* Klik op *Create tag* linksonderin.

### Stap 21:

Als je klaar bent met de ticket zet die dan in de kolom “Review”.

### Stap 22:

Schiet een gesprek in met je leidinggevende en bespreek hoe je de ticket (issue) hebt opgelost en welke gevolgen dit heeft voor de applicatie. Ook vertel je welke veranderingen er zijn gemaakt ten opzichte van de documentatie.

### Stap 23:

Zet de ticket (issue) op closed en herhaal alle stappen vanaf **Stap 17** totdat alle tickets (issues) zijn opgelost.

### Stap 24:

Als alle tickets (issues) zijn opgelost kijk je of alle applicatiedocumentatie goed is bijgewerkt. Je plant een gesprek in met je leidinggevende en vraagt of de applicatie klaar is voor een major release. Doe dit op de volgende manier:

* Verander alle bestanden van het project naar versie 2.0
* Maak een tag aan met de versie 2.0
* Commit “Mesage: major release versie 2.0”
* Push al je bestanden naar git

# Opmerkingen

Niet alle bestanden hoeven van versie te veranderen bij een aanpassing bijvoorbeeld:

* Repository
  + Project 1.1
  + Programma van eisen 1.0
  + Plan van aanpak 1.0
  + Functioneel ontwerp 1.1
  + Technisch ontwerp 1.1

Etc..

Je hoeft niet alle bestanden naar 2.0 te zetten bij de major release alleen de tag moet worden aangemaakt met 2.0 en een message met alle veranderingen.

Je hoeft niet naar iedere opgeloste issue een gesprek in te plannen met de leidinggevende alleen als iets onduidelijk is.

**Let op! De eerste stappen van het stappenplan zijn niet van toepassing omdat de git repository op het examen al is aangemaakt en in die repository moet gewerkt worden!!!**

Voor de naamgeving van de labels moet de opdracht goed worden doorgelezen en naar de bijlage gekeken moeten worden.