

Criteria en matrix versiebeheer 2025-26

Beschrijving

Om versiebeheer te kunnen beoordelen moeten de studenten 2 dingen inleveren/laten zien 1 toets en 1 assessment gesprek.

Opleveringen:

- Een eigen repository waarin ze laten zien dat ze zelfstandig de vaardigheden hebben om versiebeheer en structuur toe te passen op een code project.
- Een groepsproject repository waarin ze laten zien hoe ze structuur kunnen brengen op een professionele wijze in groepsverband doormiddel van GitHub.

Toets: een theoretische toets die een overzicht geeft van hoever je kennis gaat. Deze toets is voorwaardelijk voor het assesment en zal een deel van de basis vormen voor wat er tijdens dat gesprek zal besproken worden.

Assessment gesprek: een 5 tot 10 minuten gesprek waarbij er vragen gesteld worden over je kennis (op basis van de toets), een eigen repository en de groeps repository

Er zitten aan deze opleveringen 3 eisen die behaald moeten worden:

1. Er moet zowel een eigen als een groepsrepository zijn om te beoordelen
2. Er moet genoeg inhoud in de repository zitten om de beoordeling te kunnen uitvoeren. Deze inhoud is de hoeveelheid commits die zijn uitgevoerd. Minstens 3 voor de eigen repository en minstens 3 per groepslid voor de groepsrepository.
3. De begeleider/ beoordelaar is als collaborator toegevoegd aan de groepsrepository. Het is wenselijk voor de begeleider/beoordelaar om toegang te hebben tot de eigen repository echter is het daar geen vereiste om als collaborator toegevoegd te zijn enkel dat het zichtbaar is voor de begeleider/beoordelaar.

Zonder te voldoen aan deze 3 eisen kan er geen beoordeling plaatsvinden en zal er gelijk onvoldoende uitkomen.

De beoordeling vindt plaats in 3 delen:

1. Doormiddel van het bestuderen van het geleverde individuele repository. Voor ieder criteria uit de individuele matrix kan er 0, 1 of 2 punten gegeven worden. Voor het individuele repository moet er 9 of meer punten uitkomen.
2. Doormiddel van het bestuderen van het geleverde groepsrepository. Voor ieder criteria uit de groepsmatrix kan er 0, 1 of 2 punten gegeven worden. Voor de groepsrepository moet er 15 of meer punten uitkomen.
3. Doormiddel van een assessment gesprek met de groep van de groepsrepository. In dit assessment wordt een deel van de kennis gecontroleerd en vragen gesteld over het proces. Zie de voorbeeld vragen onderin.

Het vak wordt afgerond als de twee opleveringen voldoende zijn en het assessment, wat ter controle is, voldoende gevonden wordt.

Leerdoelen

- A. Aan het einde van het traject van versiebeheer kan de student een GitHub account maken.
- B. Aan het einde van het traject van versiebeheer kan de student meerdere repositories opzetten met duidelijke titels en beschrijvingen en weet waarom duidelijke titels en beschrijvingen belangrijk zijn.
- C. Aan het einde van het traject van versiebeheer kan de student meerdere commits met nette en consistente namen toevoegen aan een repository en kan aangeven waarom dit belangrijk is.
- D. Aan het einde van het traject van versiebeheer weet de student op een wijze de geschiedenis de repository terug te halen.
- E. Aan het einde van het traject van versiebeheer kan de student groepsleden toevoegen aan een repository.
- F. Aan het einde van het traject van versiebeheer kent de student de basis 3 branching structuur (of een andere vorm van gitflow) en kan deze toepassen en onderhouden
- G. Aan het einde van het traject van versiebeheer kan de student zelfstandig branches aanmaken en weer mergen.
- H. Aan het einde van het traject van versiebeheer kan een student de afspraken die zijn gemaakt rondom het consistent en professioneel benamen van commits, documenten en branches aan de hand van een eigen opgestelde lijst met afspraken beargumenteren en verantwoorden.
- I. Aan het einde van het traject van versiebeheer kan een student in groepsverband een GitHub project opzetten, koppelen en onderhouden doormiddel van issues en een projectonderdeel zoals KanBan of Teamoverzicht.
- J. Aan het einde van het traject van versiebeheer weet de student gebruik te maken van rechten en zichtbaarheden om de kwaliteit en veiligheid van een repository en groepsleden te waarborgen.
- K. Aan het einde van het traject van versiebeheer kan een student in groepsverband een repository aanmaken en bijhouden.
- L. Aan het einde van het traject van versiebeheer kan de student samenwerken op Github en onderbouwen welke keuzes zijn gemaakt.
- M. Aan het einde van het traject van versiebeheer kan de student evalueren over het project en over hun gebruik van GitHub.

Toetsmatrijs

X = individueel , Y = groeps, Z = assessment

Leerdoelen	Lesstof	Percentage of punten per bloom-niveau						# punten	Gewicht
		Weten	Begrijpen	Toepassen	Analyseren	Evalueren	Creëren		
A	HO1			XY(1)				1	2%
B	HO1		XZ(2)	XY(1)				3	6%
C	HO1	Z(0)	Z(1)	XY(2)				3	6%
D	HO2	Z(1)	Z(1)	Z(1)				3	6%
E	HO2			XY(3)				3	6%
F	HO3			XY(2)	Y(2)	Y(2)		6	12%
G	HO3 HO4			X(2)	Y(2)	Y(1)	Y(1)	6	12%
H	HO1 HO2 HO3		Y(1)	Y(2)	YZ(2)	YZ(1)	Y(0)	6	12%
I	HO5		Y(0)	Y(1)	YZ(2)	Z(0)		3	6%
J	HO1 HO2 HO5	Y(0)	Y(1)	Y(1)	Y(1)			3	6%
K	HO2 HO3 HO4 HO5			Y(2)	Y(1)			3	6%
L	HO3 HO4 HO5		Z(0)	Z(1)	Z(2)	Z(3)		6	12%
M	HO3 HO5				Z(2)	Z(2)		4	8%
TOTAAL		1	6	19	14	9	1	50	100%

Matrix individuele repository

Dikgedrukt = Vereist anders direct onvoldoende

	Onvoldoende (+0)	Voldoende (+1)	Uitstekend (+2)	Behaalde punten:
Het maken van repositories	Heeft geen repositories gemaakt	Heeft minstens 1 eigen repository	Heeft meerdere repositories	
README of omschrijving	Heeft geen Nuttige README of omschrijving	Heeft een korte README of Omschrijving die het doel van de repository of code omschrijft	Heeft een uitgebreide README of omschrijving waarin duidelijk staat omschreven wat de doel en werking is van de repository en code.	
Gitignore	Bevat geen Gitignore	Bevat gegenereerde Gitignore	X	
Nuttige Commits (Nuttig wordt gedefinieerd aan de inhoudelijke toevoeging van code aan een repository)	Bevat geen (nuttige) commits of minder dan 3 commits	Bevat tussen de 3 en 5 nuttige commits	Bevat 6 of meer nuttige commits	
Branching	Bevat geen vorm van branching of houd zich niet aan basis 3 branchingstructuur.	Bevat en onderhoudt het merendeel van basis 3 branchingstructuur.	Bevat en onderhoudt een goede basis 3 branching structuur of een uitgebreide vorm van de basis 3 branchingstructuur.	
Benaming (Ongepast taalgebruik wordt gedefinieerd als scheldwoorden, discriminerende woorden of taalgebruik dat tegen het beleid is van het Nova College)	Heeft geen consistente vorm van benaming van commits of branches of benamingen bevatten ongepast taalgebruik.	Benaming van commits en branches zijn voor het merendeel consistent.	Benaming van commits en branches zijn consistent en volgens eigen opgestelde naamgevingsregels.	
Merge (Nuttig wordt gedefinieerd aan de inhoudelijke toevoeging van code aan een repository)	Heeft geen (nuttige) merge uitgevoerd	Heeft een nuttige merge uitgevoerd.	Heeft meerdere nuttige merges uitgevoerd volgens branching basis 3 branching regels.	
History	Kan geen voorgaande versie of commit terughalen.	Kan een voorgaande versie en commit terughalen met wat zoeken.	Kan gericht een voorgaande versie en commit terughalen.	
Samenwerking Dit wordt getoetst aan de hand van of de beoordelaar toegevoegd is aan de repository	Kan geen collaborator toevoegen aan een repository	Kan een Collaborator toevoegen aan een repository	Kan een collaborator toevoegen en insight gebruiken om samenwerking te laten zien.	

Matrix groepsrepository

Dikgedrukt = Vereist anders direct onvoldoende

	Onvoldoende	Voldoende	Uitstekend	Behaalde punten:
Groepsrepository	Is niet aanwezig	Is aanwezig	X	
Begeleider/ beoordelaar toegang	De begeleider of beoordelaar heeft geen toegang tot de repository	De begeleider heeft toegang tot de repository	X	
README of omschrijving	Heeft geen Nuttige README of omschrijving	Heeft een korte README of Omschrijving die het doel van de repository of code omschrijft	Heeft een uitgebreide README of omschrijving waarin duidelijk staat omschreven wat de doel en werking is van de repository en code.	
Gitignore	Bevat geen Gitignore	Bevat gegenereerde Gitignore	X	
Nuttige Commits (Nuttig wordt gedefinieerd aan de inhoudelijke toevoeging van code aan een repository)	Bevat geen (nuttige) commits of minder dan 3 commits	Bevat tussen de 3 en 5 nuttige commits	Bevat 6 of meer nuttige commits	
Mappen structuur (Nuttig wordt gedefinieerd als bevorderlijk voor het vinden van een specifieke file)	Alle documenten op staan door elkaar op de branches	Het merendeel van de documenten is onderverdeeld in een nuttige mappenstructuur	De documenten en mappen structuur is netjes, nuttig en doordacht opgezet waardoor files zoeken gemakkelijk gaat.	
Branching 1	Bevat geen vorm van branching of houd zich niet aan basis 3 branchingstructuur of er is geen documentatie branch	Bevat en onderhoudt het merendeel van basis 3 branchingstructuur en heeft een documentatie branch.	Bevat en onderhoudt een goede basis 3 branching structuur of een uitgebreide vorm van de basis 3 branchingstructuur en bevat een nette documentatie branch.	
Branching 2 (Nuttig wordt gedefinieerd aan de hand van of de branch een toevoegend doeleinde heeft aan de code van de repository)	Groepslid heeft geen (nuttige) branch gemaakt in de repository	Groepslid heeft 1 nuttige branch gemaakt	Groepslid heeft meerdere nuttige branches gemaakt.	
Benaming (Ongepast taalgebruik wordt gedefinieerd als scheldwoorden, discriminerende woorden of taalgebruik dat tegen het beleid is van het Nova College)	Heeft geen consistente vorm van benaming van commits of branches of benamingen bevatten ongepast taalgebruik.	Benaming van commits en branches zijn voor het merendeel consistent.	Benaming van commits en branches zijn consistent en volgens eigen opgestelde naamgevingsregels.	

Merge (Nuttig wordt gedefinieerd aan de inhoudelijke toevoeging van code aan de repository)	Heeft geen (nuttige) merge uitgevoerd	Heeft een nuttige merge uitgevoerd.	Heeft meerdere nuttige merges uitgevoerd volgens branching basis 3 branching regels.	
Samenwerking Dit wordt getoetst aan de hand van de inzichten tab.	Groepslid heeft geen of minder dan 20 regels aan code toegevoegd aan de repository.	Groepslid heeft 20 of meer regels code toegevoegd aan de repository.	Groepslid heeft meer dan 100 regels code toegevoegd aan de repository.	
Projecten	Er is geen project aanwezig of het project is leeg of de begeleider/beoordelaar is niet toegevoegd hieraan.	Het project is aanwezig, bevat enkele nuttige items en de begeleider is hieraan toegevoegd	Het project is aanwezig, bevat vele nuttige items waaruit blijkt dat het onderhouden wordt en de begeleider is hieraan toegevoegd	
Issues (Nuttig wordt hier gedefinieerd als of het een toevoeging is aan het proces of product van de repository.)	Groepslid heeft geen (Nuttige) Issue geschreven.	Groepslid heeft minstens 1 nuttige issue geschreven.	Groepslid heeft meerdere nuttige issues geschreven.	
Taal	Er worden meerdere talen door elkaar gebruikt	Er wordt voornamelijk 1 taal gebruikt	Er wordt 1 taal gebruikt buiten vaktermen om.	
Veiligheid van project	Iedereen heeft Admin toegang of de beoordelaar/begeleider is niet toegevoegd.	Iedereen (inclusief de begeleider/beoordelaar) heeft write rechten en is toegevoegd.	Het project is privé ongeacht of de repository publiek of privé is, de beoordelaar/begeleider heeft alleen read rechten, en de rest heeft write rechten.	

Assessment voorbeeld vragen

Basis info

- Kan je een voorbeeld geven van hoe je een geschiedenis kan ophalen?
- Wat kan de inhoud zijn van een repository?
- Wat is het nut van een gitignore?
- Wat is een reden om versiebeheer toe te passen?

Branching en merging

- Welke 3 branches zijn noodzakelijk voor de basis 3 branching structuur?
- Wat zou een reden zijn om een branch aan te maken?
- Wanneer zou je kunnen mergen? Noem 1 reden
- Wat is een andere naam van de heilige branch?

Structuur/project

- Noem 3 onderwerpen die in de README zouden kunnen staan.
- Welk vorm van project hebben jullie gekozen voor jullie pygame project? -> waarom / bevalt het? ->waarom
- Voor ieder groepslid, noem een reden om een issue aan te maken.
- Wat is een van de grootste redenen om issues te gebruiken ten opzichte van losse chatberichten er zijn er 3 in totaal.
- Werkte het gekozen project voor jullie groep? -> waarom wel/niet?

Samenwerking

- Hebben jullie afspraken gemaakt over commits? -> Zo ja welke? / Zo nee waarom?
- Hebben jullie fouten gemaakt gedurende het project op het gebied van versiebeheer? -> zo ja welke? -> wat heb je geleerd van die fouten?
- Waren jullie problemen tegen gekomen tijdens het project? Zo ja welke afspraken zou je volgende keer maken om deze te voorkomen?

Ervaring/proces

- Vond je de projects handig? -> waarom wel/niet?
- Ga je versiebeheer vaker toepassen? Zo ja hoe?