

# Beoordelingscriteria voor ingeleverd werk CSD2

## Noodzakelijke voorwaarden

Code en documentatie worden ingeleverd via Git. De structuur hiervoor is beschreven in

[https://csd.hku.nl/csd2/now/csd2a/lesmateriaal/L&S1\\_HO\\_2\\_GIT\\_intro.pdf](https://csd.hku.nl/csd2/now/csd2a/lesmateriaal/L&S1_HO_2_GIT_intro.pdf)

Opdrachten die niet aan dit format voldoen kunnen niet worden beoordeeld.

Uit de commit history van je git repository moet blijken dat je git in je workflow geïntegreerd hebt.

Code moet begrijpelijk zijn voor anderen, wat impliceert dat de structuur duidelijk moet zijn, er voldoende comments in staan en dat het gesteund wordt door documentatie zoals een ontwerp, class-diagrammen en systeemdiagrammen.

Slecht leesbare code, code zonder relevante comments, slecht gestructureerde code en code met niet-verklarende namen voor files, variabelen, classes en functies wordt niet geaccepteerd.

Code die niet gerelateerd lijkt aan het ontwerp en de documentatie wordt niet geaccepteerd. Uit de documentatie moet de werking van de code al duidelijk zijn.

Niet accepteren van code betekent dat het ingeleverde niet wordt nagekeken. Na correctie voor de gestelde deadline kan het alsnog worden beoordeeld. Stel de docenten wel even op de hoogte als je op hun verzoek aanpassingen hebt gedaan.

## Beoordelingscriteria

- Ontwerp goedgekeurd.
- Het ontwerp als uitgangspunt genomen. Je mag gaandeweg best tot nieuwe inzichten komen, maar verantwoord dan wel je keuzes.
- Overzichtelijke code en betekenisvolle namen voor classes, members en variabelen.
- Zinvol commentaar in de code.
- Geen 'uitgecommente' irrelevante code
- Robuustheid van het programma: het programma moet niet crashen of onjuist werken als de gebruiker er verkeerd mee omgaat, zoals bijvoorbeeld bij het invoeren van tekst als een getal verwacht wordt of het opgeven van bestanden die niet bestaan. In zo'n geval moet het programma een foutmelding geven en/of aangeven wat de bedoeling is.
- Klinkend resultaat.

- Git geïntegreerd in je workflow: commit bij elke significante wijziging je code in git (met add/commit) en zinvol commentaar. De 'commit history' moet een duidelijk beeld geven van het ontwikkelproces.
- Duidelijke gebruikers-interactie (*indien van toepassing*). Help de gebruiker bij het gebruik van het programma. Leid de gebruiker als het ware door het programma heen.