MUDE assessment

Final grade based on 3 components:

- 2 exams (25% each)
- Assignment portfolio (50%) : weekly programming assignments + group assignment

In Delft:

- Each year 2 opportunities for each component
- Assignments (resit only if you submitted something), options:
  - If not passed, a repair option is given, and final grade is max. 6.0.
    For each question: reflect on what was wrong, and how it should have been done.
  - New assignment.
  - Hybrid.

Group assignments

How to set a deadline? How strict?

- Advantage of the 'pressure cooker' approach: fair and consistent

How to make sure everyone is contributing

- Design such that different skills / knowledge can be exploited
  > difficult since most assignments require step-wise solution (do this, then that, …)

- Promote that they explain to each other and review each other's work (ask for review report)

- Peer and self assessment

Assessing individual contributions

- Use peer review (e.g., feedbackfruits) and peer- and self-assessment (e.g., buddycheck). Peer review between groups or within groups: Students must review each other's work and submit review report

- Authentication report as deliverable: all team members indicate which parts they (co-)authored, and which parts they reviewed (may include review report)

- Self-reflection on learning objectives

- Each team member must submit own version of introduction (at start) and/or of the conclusions and recommendations (at end)

- Scrums with one or two team members at a time during the project

Grading of weekly MUDE assignments:

Options:

- Simple rubric based on set of general criteria, individual questions not graded > requirement is that you submit solutions
- Students need to write a short report, answer questions (more on interpretation) related to the notebook
  - answers are graded based on answer model
  - in addition: some additional checks on quality of notebook (visuals, coding standards

General criteria - examples

Documentation, use of metrics, visualization
Application: More than X % of questions seriously answered
Programming: Applied proper coding standards,
Interpretation: Show clear understanding and insights

Category	No Submission	Insufficient	Sufficient	Very good	Default Weight
Documentation	0	1	3	5	**0.15**
Application	0	1	3	5	**0.15**
Programming	0	1	3	5	**0.15**
Interpretation	0	1	3	5	**0.55**

Note: it is advisable to introduce an additional 'level' for 'good'

Grading scheme for assignments

- 2.5 points for each Programming Assignment after *successful* submission

- max. 20 points for Project