EdTech: CodeGrade Evaluation

Seyyed Kazem Banihashem, Omid Noroozi

Education and Learning Sciences (ELS)

Wageningen University and Research (WUR)

Evaluation team

Seyyed Kazem Banihashem

Postdoctoral Researcher (ELS-WUR)

Main evaluator

Omid Noroozi

Associate Professor (ELS-WUR)

Supervisor

We acknowledge the valuable contributions and feedback of Prof. dr. Perry den Brok and Dr. Harm Biemans on this report.

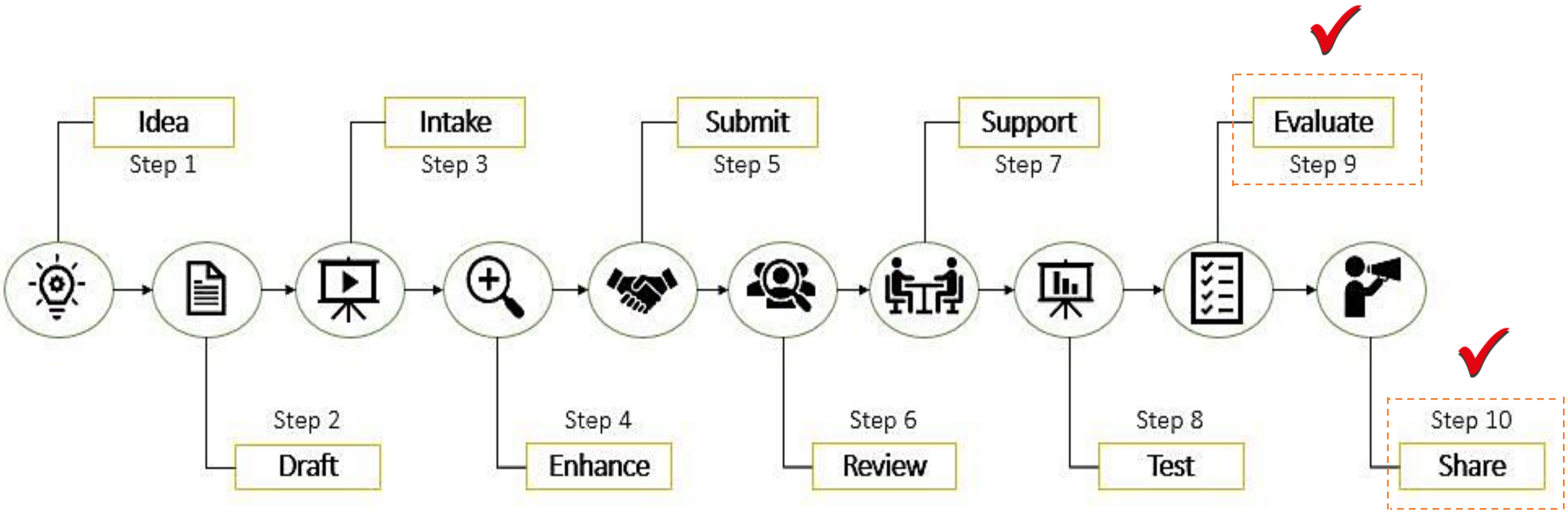
Content

- EdTech community
- Overview of CodeGrade
- Method
- Key results
  - Teachers’ view
  - Students’ view
- Conclusions
- Recommendations

EdTech Community

EdTech community

- Supporting teachers who experiment with new EdTech tools in education
- Enhancing quality of education at WUR with EdTech tools
- EdTech evaluation: Step 9 (evaluate) and Step 10 (share)



Overview of CodeGrade

Overview of CodeGrade

- A grading and feedback platform for computer science
- Goals:
  - Maximize computing educators’ teaching performance
  - Help teachers with grading and feedback on codes
  - Help students to improve coding skills
- Some functionalities:
  - Plagiarism detection
  - Auto grading
  - Feedback
- Website: https://www.codegrade.com/

Methodology

Context

- Course A
  - CodeGrade was mandatory to be used
- Course B
  - CodeGrade was mandatory to be used
- Course C
  - CodeGrade was optional to be used
- Course D
  - CodeGrade was mandatory to be used

Context

- Course A

CodeGrade was used for submissions and providing feedback and giving grade.

Course detail

Language of instruction:

- English

Level of instruction:

- MSc

Number of students:

- 90 Students

Period:

- Period 2 (CodeGrade Tested)

Course goal

Some course goals:

- Design and implementation of a bioinformatics analysis pipeline
- Applying and automating advanced batch-wise sequence searching and analysis techniques

Course evaluation

- Written reports on the results obtained with the exercises (30%)
- A written report on the research project (50%)
- An oral presentation of the research project (20%)

Context

- Course B

CodeGrade was used for some deliverables of the course such as weekly assignments, an exam, and a final project.

Course detail

Language of instruction:

- English

Level of instruction:

- MSc

Number of students:

- 70 Students

Period:

- Period 3 (CodeGrade Tested)

Course goal

Some course goals:

- Understand basic concepts of applied scripting for spatial data
- Read, write, and visualize spatial data (vector/raster) using a script
- Apply learned scripting concepts in a case study with geo-data

Course evaluation

- Daily handed in assignments (30 %)
- Final week assignment/ project (60%)
- Participation in the course online forum (10%)

Context

- Course C

CodeGrade only was used for self-assessment as an extra practice and as a bonus.

Course detail

Language of instruction:
- English

Level of instruction:
- MSc

Number of students:
- 50 Students

Period:
- Period 5 (CodeGrade Tested)

Course goal

Some course goals:
- Implement a given algorithm as a computer program (in Python)
- Explain what a given piece of programming code (in Python) does
- Identify and repair coding errors in a given piece of programming code

Course evaluation

- Written closed book exam

Context

- Course D

CodeGrade was used for peer feedback experiment.

Course detail

Language of instruction:

- English

Level of instruction:

- MSc

Number of students:

- 34 Students

Period:

- Period 6 (CodeGrade Tested)

Course goal

Some course goals:

- Identify the algorithmic design technique and the underlying optimization criterion in bioinformatics algorithms
- Implement a given bioinformatics algorithm in python

Course evaluation

- Reports handed in on the practical assignments (60%)
- Literature presentations/ discussions given (30%)
- Participation (10%)

Participants

Data from teachers – Focus group discussions

- Course A (N=3)
- Course B (N=4)
- Course C (N=2)
- Course D (N=3)
- Analysis of demo on CodeGrade - Course B (Recoded Video - P3)

Data from students – Students' satisfaction [PACE] and Interview with students

- Students' satisfaction [PaCE] data from Course B (N=45)
- Students' satisfaction [PaCE] data from Course C (N=20)
- In-depth interview with students – Course C (N=3)
- Focus group discussions with students – Course D (N=2)

Instruments

- Open-ended questions for teachers:

1. Are you generally satisfied with using CodeGrade? Why No/Yes? Rate from 1 to 10.
2. To what extent CodeGrade was successful in achieving learning goals?
3. Did CodeGrade add any other values to the course?
4. Did CodeGrade help you to decrease your workload?
5. How would you explain ease of use and enjoyability of CodeGrade?
6. From your point of view, what were the weaknesses and the strengths of CodeGrade?
7. Do you have any suggestions for further improvements of CodeGrade?
8. Would you recommend using CodeGrade for other similar courses?

Instruments

- Open-ended questions for students:

1. Are you generally satisfied with using CodeGrade? Rate from 1 to 10.
2. Did CodeGrade help you to achieve learning goals?
3. Did CodeGrade help you to decrease your workload?
4. How would you explain ease of use and enjoyability of CodeGrade?
5. From your point of view, what were the weaknesses and the strengths of CodeGrade?
6. Do you have any suggestions for further improvements of CodeGrade?
7. As a student, would you recommend using CodeGrade for other similar courses?

Note: We were not able to conduct interview with students in the courses A and B due to absence of students' participation.

Instruments

- CodeGrade statements for students’ satisfaction [PaCE] in Course B

1. I liked the deliverable submission process in CodeGrade.
2. The feedback I received from CodeGrade contributed to my learning.

✓ Type of rating: Likert-Scale from strongly disagree (1) to strongly agree (5)

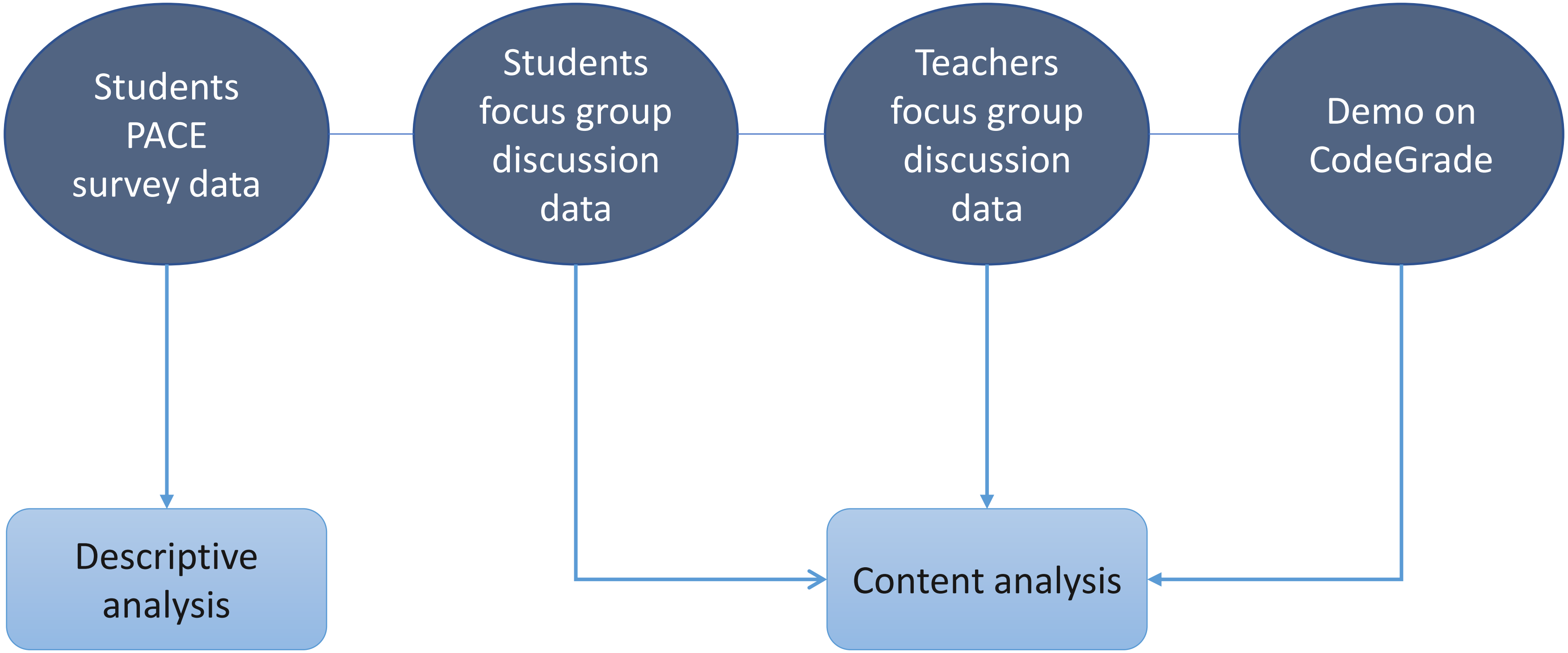
- CodeGrade statements for students’ satisfaction [PaCE] in Course C

1. Self-tests using CodeGrade contributed to the development of my skills.
2. CodeGrade was useful for submitting self-test assignments and receiving scores.

✓ Type of rating: Likert-Scale from strongly disagree (1) to strongly agree (5)

Note: We do not have PaCE evaluation results of students in the courses A and D due to missing the deadline which was out of our control.

Data analysis

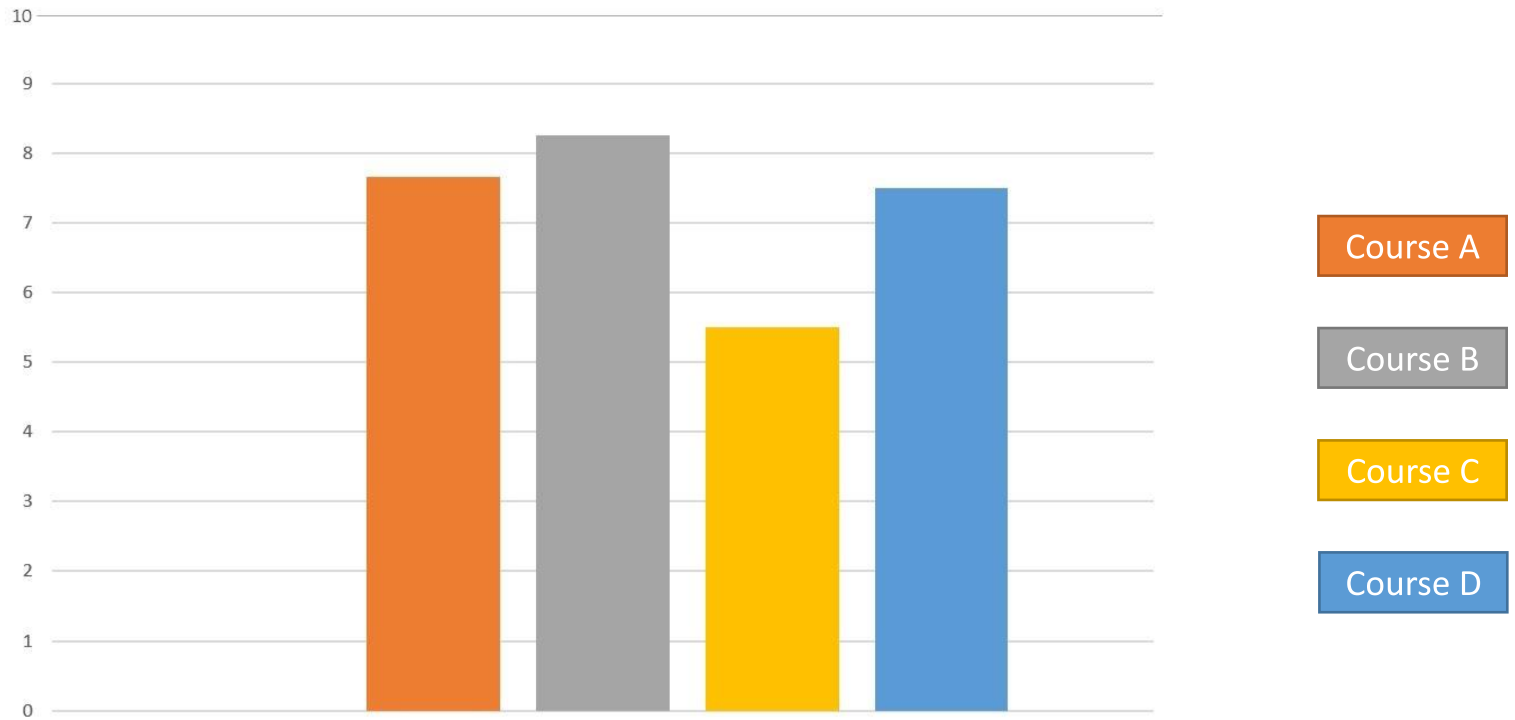

Students PACE survey data
Students focus group discussion data
Teachers focus group discussion data
Demo on CodeGrade
Descriptive analysis
Content analysis

Note: Data collected through CodeGrade's demo has provided us with concrete information regarding the tool's features, potentials, and functions.

Key results

Teachers

General satisfaction (teachers)



- *“I think that this tool makes the life of the teaching stuff much easier, and I will give 8 out of 10.”*
- *Yes, for me it was also convenient compared to the feedback I got last year for this course.*

Learning goals (teachers)

Course A (helpful)

- **Providing feedback** *“I do think that we normally give feedback as well but now with CodeGrade you can give more directive feedback on piece of code. So, I think in terms of providing feedback on students’ work as a learning goal, it already added some benefits I would say, because for example with CodeGrade you could type immediately through the lines.”*

Course B (helpful)

- **Peer feedback** *“For me I think there is a feature to peer group feedback which could contribute more to the learning goals. CodeGrade can let students to provide maybe more in detailed feedback or line by line feedback.”*
- **Self-practice and feedback** *“I think the auto testing and giving feedback immediately to students whether they are right or wrong helped students, and it helped students to practice with not necessarily presence of teachers. So, I think CodeGrade generally helped with achieving the learning goals of the course.”*

Learning goals (teachers)

Course C (helpful)

- **Monitor the process** *"In terms of the learning goals, what we wanted it was to monitor the process, and I think it was useful in that sense."*
- *"But keep it in mind that this was a small course that we tested CodeGrade. So, this positive impacts on achieving the learning goals that we expected from CodeGrade (monitor progress, auto grade) need to be confirmed by our other courses which we have more than 200 or 300 students."*

Course D

- ---

Added values (teachers)

- Course A (negative)

- *"For the future I think there could be a lot of possibilities because we have not used it extensively but at the moment no."*
- *"I don't think it added other values a lot yet. It just helped us to produce new ideas based on its potential for the course."*

- Course B (positive)

- **Force to a good practice** *"I think the tool forces you to a good practice."*
- **Self-practice** *"It could bring added values for self-practice of students."*

Added values (teachers)

- Course C (negative)

- *“Not that I can think of to be honest.”*
- *“It is always nice to use a new app, but I think in that sense it did not add that much.”*

- Course D

- ---

Workload (teachers)

Course A (slightly increased workload)

- **Setting up the auto-test** *“I think setting up the auto test and those kind of things still took a lot of my time.”*
- *“If we compared it with what we did last year, I don’t think it reduced it much. I don’t think it decreased workload.”*

Course B (decreased workload)

- **Grading** *“The grading was easy because you had rubrics and you just need to click on. For example, this is five, this is six point, and so on. Each rubric had a description what exactly it means.”*
- **Group assignment** *“What that really helped with time saving that you can split up the tasks to the people that you designate.”*

Workload (teachers)

Course C (increased workload)

- **Technical issues and students' emails** *"Maybe you can ask the other way around. This is the main critic. Because it really increased the workload on teaching assistants, and teachers themselves. Because, we had to also respond students emails about CodeGrade and solve the problems."*

Course D (decreased workload)

- **Automation function** *"Well, I did not dive much into CodeGrade but I think the automation in CodeGrade could save time for me."*

Ease of the use and enjoyability (teachers)

Course A (not strongly user friendly)

- **Too many buttons** *“If definitely took some time to get used to. It had quite a lot of buttons you need to figure out what were they for.”*
- **Small window** *“Well, I liked the features, but I think the interface was not sometimes optimal. The window that you can see the script was quite small.”*

Course B (Slightly user friendly)

- **Simplified for students** *“I think it was very easy to use in general. For students, the interface was very simplified and very straightforward where to get feedback.”*
- **Icons were not obvious** *“It was easy to use but it is true that for example some icons were there, but it was not obvious. So, all the functionalities were there, and it was a good point, but you did not realize that you need to click on that.”*

Ease of the use and enjoyability (teachers)

Course C (User friendly)

- *“For me, it was quite user friendly. For example, if I remember correctly, there was a screen with all the students’ name, and it was quite handy to check each student.”*
- *“I also think it was easy to use. We did not receive any complaints from students about difficulty in dealing with CodeGrade.”*

Course D (Not strongly user friendly)

- *“For me as well, it was relatively straightforward to use.”*
- *“I did have quite a bit of trouble getting everything sets up in CodeGrade. Right now, I still do not know how to copy my CodeGrade assignment in Brightspace.”*

Weaknesses (teachers)

- **Interactive scripting “***There is a mismatch between what we do now and what CodeGrade does. For example, you cannot run the code interactively. So, you can not change their line and run it again to see whether the error was there or not.”*
- **Transparency in grading** *“One of the weaknesses maybe is that the auto grading process is not transparent. It did not dig into what is wrong and why this point is given to the students.”*
- **Technical issues and missing features** *“There are still some bugs, technical issues, and missing features.”*
- **Poor functionality** *“The number of functionalities were quite acceptable, but the functions did not work very efficiently.”*
- **Auto grading’s reliability issues** *“Maybe in 40 or 50 percent of cases, the auto-grading results were not reliable.”*
- **Inflexibility** *“When students use alternative methods, the app did not recognize it as correct.”*
- **Increased workload** *“For example, teaching assistants had to check the grade and results manually for maybe 40 or 50 percent of the students.”*
- **Small window** *“When you want to write something the box or window was rather small, and it was kind of annoying specially if you are writing a big piece of feedback.”*
- **Not supporting word file** *“For students who have uploaded the word file instead of pdf file, I was not able to directly add comments to the word files. The word file even did not open.”*

Strengths (teachers)

- **Direct feedback** *"I think, one of the strengths is that you can give the feedback directly in the tool that you can attach the feedback to either a function or a specific line."*
- **Auto grading** *"Also the potential strength is the auto grading and input-output check that you can do."*
- **Plagiarism check** *"There is also a plagiarism check in the tool which is convenient because, coding scripts work differently than normal texts."*
- **Rubric availability** *"One other strength was the rubric at the bottom. You could scroll to script and feedback and always could see the rubric at the bottom. I think that was really nice."*
- **Time saving** *"It helped us to save a lot of time."*
- **Providing structure** *"It offered a skeleton scaffold for what we were doing. It kept us on track with providing high quality course and at the same time being easier for us to do it.*
- **Self-assessment** *"I think that the program gave students a chance to do the self-assessment."*
- **Interface** *"Interface was fine, and I think it was a strength for the tool."*

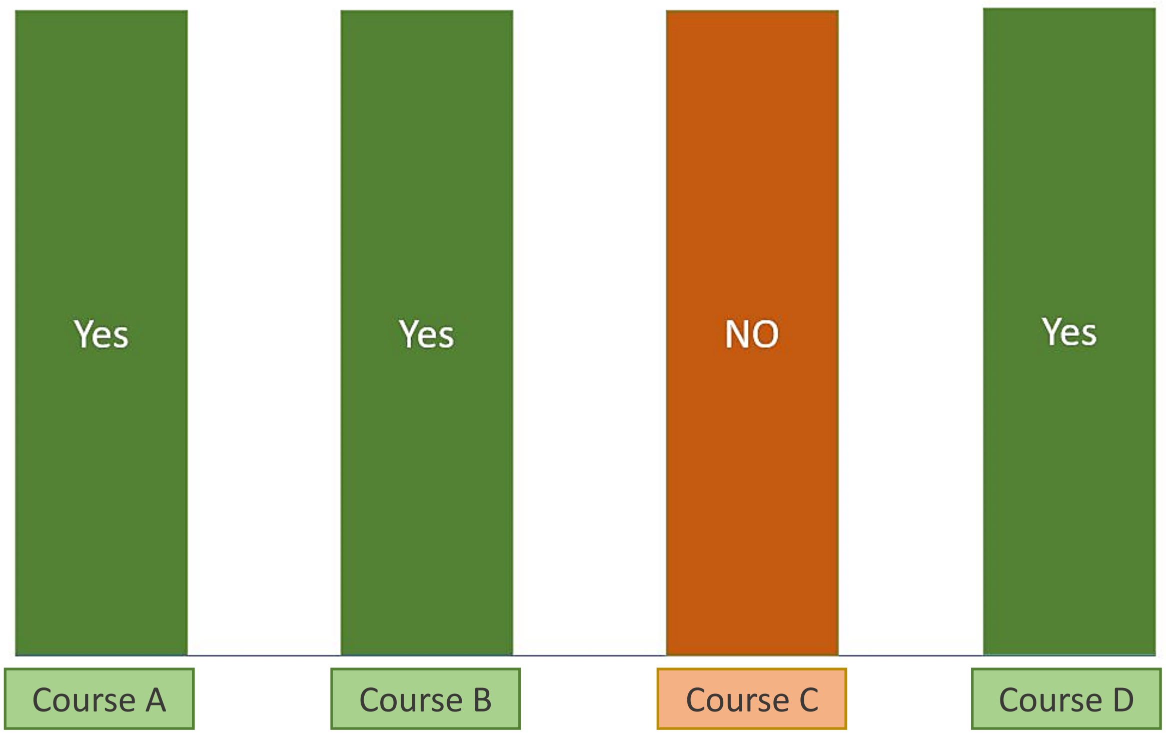
Suggestions for further improvements (teachers)

- **Book download** *“Downloading all of the scripts. So, book download of all the submissions.”*
- **Peer assessment** *“Peer assessment addition to the peer feedback.”*
- **Interface friendliness** *“Making the interface more friendly. Such as solving the small windows or other issues we discussed.”*
- **Variations in marking students** *“Apart from the percentage kind of grading, give an option to give more concrete points which is more intuitive for students. For example, now, if a student gets 2.5 out of 3 points, you need to fill in 83.3 % instead of just giving 2.5 point.”*
- **Activate “general feedback” button for students** *“last point is that some students accidently uploaded their script and report files in a zip file, instead of two separate files. So, I told the students to download the code and type the feedback under "General feedback", but it turned out the students don't have access to this button.”*
- **More flexibility in grading** *“Have more flexibility in the grading or rubric itself. For example, we would like to see that auto grading can be adapted by teachers’ rubric. Also, I think if more tests of the different types of codes were adapted by teachers’ rubric, then the program can become more flexible in grading students code.”*
- **Search icon** *“The search icon would be nice to check the submissions for students if it is not already there.”*

Suggestions for further improvements (teachers)

- **Grade announcement** *“it would be better if there was a link to the grade or a sort of announcement, once the grade is available. Because, when I released the grade, no one saw that, and I did not put an announcement because I thought that they would see it automatically.”*
- **Support word and pdf document** *“It would be also useful if word documents can also be supported because we were not able to give feedback directly on pdf or word document.”*
- **Monitor progress** *“It would be helpful if it could be seen somewhere to check if people are doing their job or not.”*
- **Deadline display** *“Under manage assignment, where I can set the duration for peer feedback, it would be helpful if the 'deadline' is displayed.”*
- **Anonymous grading** *“CodeGrade currently shows who has graded the assignment, which may not be the information you want to give to your students. Perhaps an option to put this to ‘anonymous’ or ‘grader’ would be better.”*
- **Transparency in grading** *“It would be nice if they could make the auto grading more transparent.”*
- **Automatic notification** *“I don’t know whether this functionality already exist in CodeGrade or not to automatically send an email to students with their grades when it is done as a quick notification instead of collecting them and putting them on Brightspace.”*

Recommendation to use (teachers)

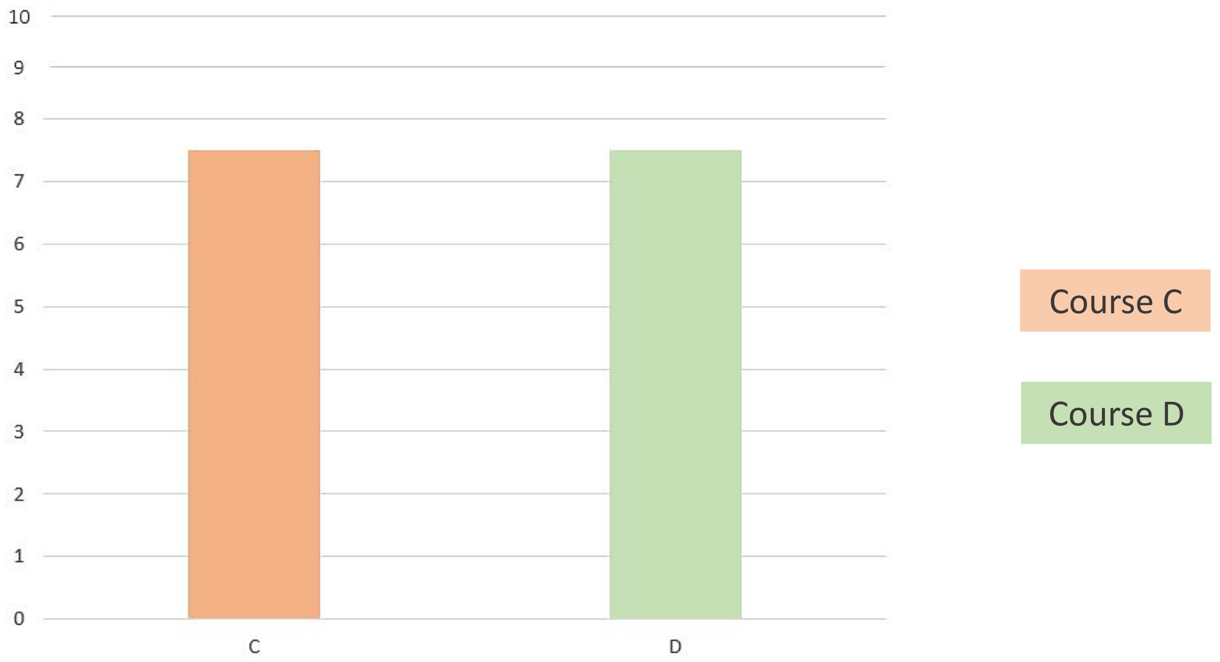


- *“I would definitely recommend using it specially in script-based courses.”* **(+)**
- *“I would also recommend using this tool for other courses that are dealing with coding.”* **(+)**
- *“Not at the moment because it added more workload, and we did not find it very useful.”* **(-)**

Key results

Students

General satisfaction (students)



- *“I think I am happy of using CodeGrade, because it was really nice that you could upload your file in the program, and you could immediately see if you did it well or not. I would give it 7.5 out of 10.”*

Learning goals (students)

Course C (slightly helpful)

- *“It was good in terms of telling what is wrong and what is right.*
- *“I think it was a good addition for learning goals. It was a different environment than Python. It had this option to see your final answer for exercises to test and to see if it was correct. So, I think it was rather helpful as a test exam.”*
- *“I would say that without CodeGrade it would have been the same. It did not help much in reaching to the learning goals.”*

Course D (slightly helpful)

- *“The only thing that I can think of is used for peer feedback once, but I don’t think it had a big impact on learning goals.”*
- *“I think it was positive, but I don’t think that it played a big role. I think if you continue using CodeGrade there could be more improvements.”*

Workload (students)

- Course C (slightly decreased workload)

- *“It was nice to get immediate feedback, so you did not have to check by yourself. In that case it would save time. But, in other cases, it doesn’t decrease the workload.”*
- *“CodeGrade saved me time by giving immediate feedback and check whether my code was correct or not. But, during the coding I did not experience any benefits in this regard.”*

- Course D (no impacts on workload)

- *“I don’t think of any impacts on saving time. Because it is normally one click in Turnitin and now another click in Turnitin.”*

Ease of use and enjoyability (students)

Course C (slightly positive)

- *“At first, I found it a little be confusing, but later I did not have any difficulties to use it.”*
- *“Yes, definitely user friendly, enjoyable and intuitive. I did not have troubles in using CodeGrade.”*

Course D (slightly positive)

- *“Yes, I think it was really intuitive for me. But I remember that at the beginning it was a bit confusing for other students to know how to get feedback, but I did not find it difficult.”*
- *“Yes, it was quite straightforward. The only problem that I had, was the small screen and that is why it was sometimes difficult to get feedback because you had to scroll up and down.”*

Weaknesses (students)

- **Not in-depth auto grading** *“Weakness is if you do not have the full points, you don’t know exactly what you did wrong, and it doesn’t say what you have to do .”*
- **Confusion in auto-grading processing** *“It was not clear at once that it was processing your answers and the window. At first, I did not realize that it was processing it, so I was uploading again.”*
- **Reliability issues in auto-grading** *“Sometimes, if there is a tiny error in your code, CodeGrade will consider the whole code wrong, while the rest of code was pretty okay.”*
- **Interface** *“The weakness I think is that you cannot really see all the feedback in a nice visual way.”*

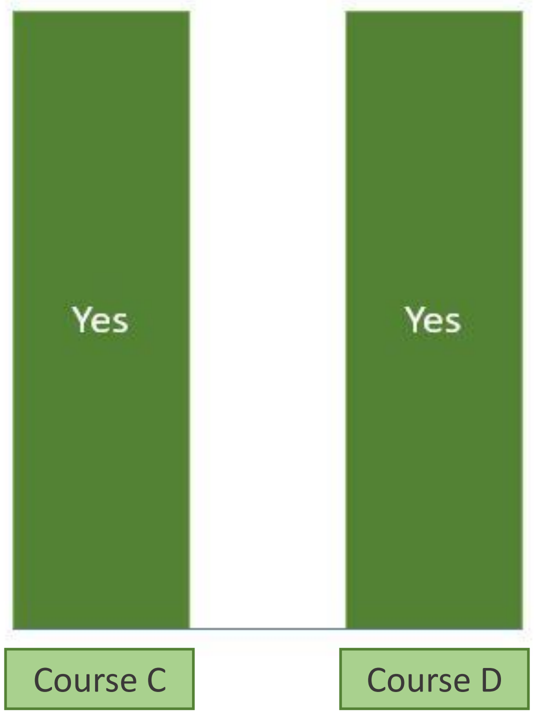
Strengths (students)

- **Immediate feedback** *“I think the biggest strength is that you can get an immediate feedback.”*
- **Easy to use** *“I think it was very easy to use which is a kind of strength for this program.”*
- **Direct feedback** *“One of the biggest strength of CodeGrade in the way that we have used it. We were able to give feedback on specific lines of the code.”*

Suggestions for further improvements (students)

- **Add an answer model** *“It’s nice to add something like an answer model. That would be nice in my opinion. I mean, if you don’t get a full point, then you would see what a correct answer should look like.”*
- **Larger window** *“I would say maybe the layout, a larger window would help.”*
- **Clear processing** *“Clear processing that you can see whether it is uploading or processing.”*
- **Possibility to type code in CodeGrade** *“I don’t remember if there was such an option to write code directly in CodeGrade or you first have to write in other initial programs like R or Python and then put it in CodeGrade. Because that would be very convenient if you could just type in CodeGrade and it would be nice.”*
- **Intelligent feedback** *“And also making feedback more intelligent maybe. For example, to divide code into small sections and say which section is correct and which section is not correct.”*
- **Bigger screen** *“Maybe one is to make the screen bigger.”*
- **Peer learning** *“I don’t know whether it is possible to see how other students do and how do they write their codes. Because it could be also nice if you could compare your code with others. Maybe it is better to see how the others solve the problems for their coding.”*

Recommendation to use (students)



- *"I have never done another course like this, but I think it could be really useful to use CodeGrade in such courses."*
- *"I think using this tool could be a nice addition for other courses."*

Students’ satisfaction [PACE] results (students)

Course B

I liked the deliverable submission process in CodeGrade

- ***N = 45 , Avg = 3.8 out of 5***

The feedback you received from CodeGrade contributed to your learning

- ***N = 45 , Avg = 3.2 out of 5***

Course C

Self-tests using CodeGrade contributed to the development of my skills

- ***N = 20 , Avg = 3.5 out of 5***

CodeGrade was useful for submitting self-test assignments and receiving scores

- ***N = 20 , Avg = 3.2 out of 5***

Conclusions

Conclusions

- Teachers and students are somewhat satisfied with using CodeGrade **(+)**
- Teachers and students recommend using CodeGrade **(+)**
- CodeGrade helps to achieve learning goals **(+)**
- Benefits of CodeGrade slightly outweigh its drawbacks **(+)**
- CodeGrade does not help much to decrease workload **(-)**
- CodeGrade is not sufficiently user friendly **(-)**
- CodeGrade needs more flexibility, interactivity, and intelligence **(-)**

Recommendations

Recommendations

- CodeGrade contributes to the quality teaching and learning in the participated courses by having features such as:
  - Auto-grading
  - Plagiarism check
  - Self-assessment
  - Structure
  - Feedback
- For large scale use, it would be great to make improvements in terms of:
  - Reliability
  - Flexibility
  - Interactivity
  - Intelligence
  - Interface

Thoughts/Questions?

Contact:

kazem.banihashem@wur.nl

omid.noroozi@wur.nl