INTEGRATING CODE REVIEW INTO HIGH SCHOOL PROGRAMMING PROJECTS

Z. Kubincová, I. Csicsolová

Comenius University in Bratislava (SLOVAKIA)

Abstract

In our study we explored the potential of peer evaluation of students' work in upper secondary education. Mutual evaluation technique was employed in informatics teaching in the form of code review – the reciprocal commenting on classmates' programming projects. Although at the beginning, the students perceived code review as an extra work, many of them later recognized its usefulness and appreciated its benefits. We believe that also thanks to mutual commenting on program codes the students developed really good projects and achieved very good grading from the teacher. Moreover, after completing this activity, the teacher observed higher self-reliance of students during development of further program codes.

Keywords: Programming, Code review, High School.

1 INTRODUCTION

The mutual evaluation of the results of classmates' work – peer review – has been used in educational activities for several years [1, 2]. Numerous studies [3, 4, 5] confirm that the peer review can provide students with various benefits, such as development of their critical and analytical thinking, interpersonal skills, constructive criticism, strengthening of social learning, etc. Apart from this it can also support their interest in the subject itself and can even help them to improve their learning outcomes.

The most common way to use peer review in teaching is to review essays or blog articles [6, 7], but there are also other commonly used implementations of this activity, depending on the subject taught. Our goal was to use mutual evaluation in informatics, thus we involved this activity in the form of a code review when teaching programming.

Code review is a technique extensively used in software companies during the development of major software projects [8] where it has proven to improve the quality of the program code and the overall effectiveness of project development [9, 10].

Using this technique in teams of professional programmers has inspired the educational professionals to use and explore it as an educational activity. Most of the published studies discussing code review in education we have found were from university courses [11, 12]. As our aim was to focus on upper secondary education, we started to carry out a study of educational activities using code review at high school. In this article we bring partial results of our study.

2 INTRODUCING CODE REVIEW IN HIGH SCHOOL TEACHING

The study was conducted under real teaching conditions in two classes of the third grade at a bilingual high school with a five-year study. It was carried out during the 2017/2018 school year in informatics classes while the *Algorithmic problem solving* (i.e. programming) unit was taught. Students learned to write code in Python programming language. The teaching of this unit took place from November to June, so the research stretched in both halves of the school year. Informatics subject was taught one teaching hour a week.

Aiming to determine the impact of employment of code review on student learning we designed activities related to this technique. During conducting our study, two types of code review activities were involved in teaching:

- **Small reviews** that had the form of a short test after certain topic was explained and practiced. In each such test, students have reviewed two short programs prepared by the teacher and written on a paper (they could not run the programs). Their first task was to find out what would the result of the first program code be if it was executed. In the second task they were asked to

find errors (syntactic and logical) in the other program and correct them. More detailed information on this activity has been published in our other paper [13].

- **Project reviews**. During the school year, students had to program two larger projects – the first one in the first half, the second one in the end of the second half of the school year.

In this paper, we focus mainly on the results of activity of the latter type – project reviews.

3 PROJECT CODE REVIEW

3.1 Project activity

Starting from the beginning of programming teaching, small reviews were employed as educational activities. Small reviews were the first opportunity for our students to experience code review. After several weeks of practicing these activities, the students were tasked to program a larger programming project which they also peer-reviewed subsequently. This activity better corresponded with real code review since students were supposed to comment on larger programming codes of their classmates.

The first project was assigned to students as homework during the winter holidays. At that time, in addition to the curriculum of the previous year, they had mastered the following topics: random colors, for-loop, functions, mouse click events, and keyboard events. The entire period for the whole project activity including commenting on the others' projects was 5 weeks.

During our research we used code review of projects twice. The second project was assigned to the end of the programming unit teaching in the second half of the school year, after explaining and practicing conditions, timer, and moving of the canvas objects. The project was noticeably more difficult than the first one as more difficult programming concepts were taught already, and students had to combine all the knowledge they gained during the programming lessons. The students asked for a longer time to program the project, so the cumulative time they spent on it was 7 weeks.

In the first project, students were supposed to choose one of the three project topic variants – car racing game, drawing application and postcard generator – so the projects of the same topic could occur several times in each group. Each of these assignments has been specified in detail.

In the second project, 17 different types of assignments were offered, such as Rock-Paper-Scissors, Lotto, Ball bouncing, Sophisticated drawing application, T-Rex, and others. Students were asked to choose one assignment, so that all projects in the same group of students were of different type. As with the first project, detailed specifications were prepared for these assignments. Apart from these topics it was possible for students to suggest their own project topic which, however, had to be approved by the teacher.

3.2 Sample

The number of students participating in the study was 52 in the first half, while in the second half there were 55 of them. The three students, who joined the classes in the second half, met with programming in the previous school year, but have not written code in the first half of this school year.

All the students were divided into four groups, A1, A2, B1, and B2. In group A1 there were 13 students (14 in the second half) of which 10 girls and 3 (4) boys; in group A2 there were 14 students of which 5 boys. In group B1 there were 12 students (13 in the second half) of which 4 boys; and in group B2 there were 13 students (14 in the second half) of which 6 boys.

Students have already programmed in the previous school year. They met with the basic graphic commands of the tkinter library, with random numbers, variables, and the for-loop. These topics were first reviewed and only afterward the teacher continued to teach new ones: functions, mouse events, keyboard events, conditions, timer, and moving of canvas objects.

Although the students learned to program in the previous school year they can still be perceived as beginners. None of them met with the code review before.

3.3 Methodology

Together with the code review of projects we introduced a possibility to improve the project based on the comments received from the classmates. In addition to giving students the opportunity to achieve better grading, we also believed this could motivate them to engage in code review activity.

All project work could be divided into three basic phases: A. Project development, B. Code review of the classmate's project, C. Customization of their own project. The particular phases involved the following steps:

a) Project development
   - Selection of one of the proposed project themes
   - Programming the project
   - Submitting the project for reviewing

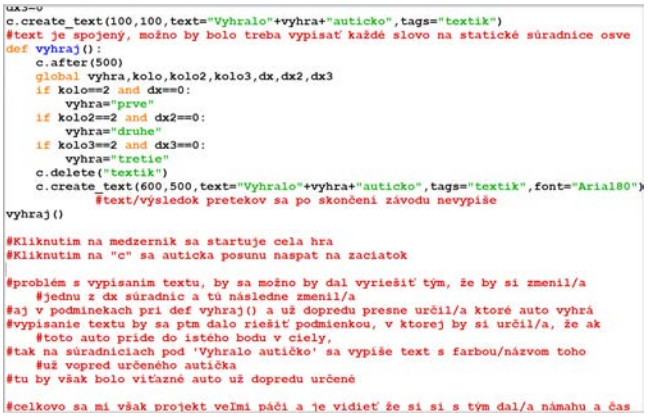
b) Reviewing a classmate's project
   - Assigning of projects for review
   - Commenting on the assigned project
   - Submitting a commented project

c) Customization of the project
   - Returning the commented project to its author
   - Modification, completion, and/or improvement of the project
   - Submitting the final version of the project for teacher evaluation

The first phase – project development – lasted three weeks for the first project, and five weeks for the second one. This extended time was not only a consequence of the higher difficulty of the latter project, it was set like this also because the students developed it alongside other school duties, while the first project was being programmed during holidays.

Since we were not able to find any tool for program distribution and code reviewing that would be suitable for high school students, the students have delivered their projects to the teacher via e-mail. The teacher assigned projects to other students for reviewing so that the author and the reviewer did not belong to the same group. This is how we tried to guarantee the anonymity when commenting on the projects – the project author was not aware of who reviewed her program, and the reviewer did not know the author of the reviewed program.

The role of a student-reviewer was to check the assigned classmates' project: to find out whether she has followed all the parts of the assignment, to identify errors and point them out, to advise a classmate, what needs to be modified or added, and write her overall opinion on the project (Figure 1). There was one week assigned for commenting on the projects. Afterwards, the students e-mailed the commented program back to the teacher who returned it to its author.



```
dx3=0
c.create_text(100,100,text="Vyhralo"+vyhra+"auticko",tags="textik")
#text je spojený, možno by bolo treba vypisať každé slovo na statické súradnice osve
def vyhraj():
    c.after(500)
    global vyhra,kolo,kolo2,kolo3,dx,dx2,dx3
    if kolo==2 and dx==0:
        vyhra="prve"
    if kolo2==2 and dx2==0:
        vyhra="druhe"
    if kolo3==2 and dx3==0:
        vyhra="tretie"
    c.delete("textik")
    c.create_text(600,500,text="Vyhralo"+vyhra+"auticko",tags="textik",font="Arial80")
            #text/výsledok pretekov sa po skončení závodu nevypíše
vyhraj()

#Kliknutim na medzernik sa startuje cela hra
#Kliknutim na "c" sa auticka posunu naspat na zaciatok

#problém s vypisaním textu, by sa možno by dal vyriešiť tým, že by si zmenil/a
    #jednu z dx súradníc a tú následne zmenil/a
#aj v podmienkach pri def vyhraj() a už dopredu presne určil/a ktoré auto vyhrá
#vypisanie textu by sa ptm dalo riešiť podmienkou, v ktorej by si určil/a, že ak
    #toto auto príde do istého bodu v ciely,
#tak na súradniciach pod 'Vyhralo autíčko' sa vypíše text s farbou/názvom toho
    #už vopred určeného autíčka
#tu by však bolo víťazné auto už dopredu určené

#celkovo sa mi však projekt veľmi páči a je vidieť že si si s tým dal/a námahu a čas
```

Figure 1. Part of a program commented by a student

In the final phase, which lasted for another week, the student's job was to incorporate the reviewer's comments on her program (those she perceived as justified), or fix the mistakes she had found by

herself in the meantime. In this way, students have been given the opportunity to correct the deficiencies in the project and improve it in order to achieve a better mark.

The teacher monitored the students' activity and the results of their work at each phase. She watched how the student managed to program the project in the first phase, how she commented on the classmate's project in the second phase, and finally evaluated the final project and checked which classmate's comments she incorporated to the program and how well she did it. After the last phase finished, she sent an individual feedback to each student, where she commented on the student's work during the individual phases of the project.

4 RESULTS

After completion of each project activity, the teacher evaluated the projects and compared the students' success rate. Afterwards, students were asked to fill in questionnaires. The questionnaires were delivered online and were filled in anonymously. The goal of this survey was to find out students' views on peer-reviewing of projects and their experience of code review activity.

4.1 Project evaluation results

For all project topics, the specific requirements that needed to be met were specified in advance. The common criteria that were taken into account in the final project evaluation by the teacher were as follows: full functionality, correctness and authenticity of the project, commenting on the classmate's project (i.e. finding errors, pointing them out, proposing solutions, overall view of the project), and customization of the own project according to the classmate's comments. For project features beyond the assignment, students earned additional bonus points. Since both projects were challenging tasks for students, they contributed more to their overall evaluation in informatics than other ordinary homework.

Table 1. Share of students with the respective percentage rating for the 1st project by groups

Group	100-90%	89-75%	74-50%	49-30%	29-1%	0%	Exceptionally good projects
A1	17.3%	5.8%	1.9%	-	-	-	7.7%
A2	21.2%	1.9%	1.9%	-	-	1.9%	9.6%
B1	11.5%	7.7%	-	-	-	3.8%	11.5%
B2	23.1%	-	1.9%	-	-	-	5.8%
Overall	73.1%	15.4%	5.8%	-	-	5.8%	34.6%

Table 2. Share of students with the respective percentage rating for the 2nd project by groups

Group	100-90%	89-75%	74-50%	49-30%	29-1%	0%	Exceptionally good projects
A1	18.2%	3.6%	1.8%	-	1.8%	-	7.3%
A2	10.9%	7.3%	5.5%	1.8%	-	-	9.1%
B1	14.5%	3.6%	-	3.6%	-	1.8%	10.9%
B2	10.9%	7.3%	3.6%	3.6%	-	-	5.5%
Overall	54.5%	21.8%	10.9%	9.1%	1.8%	1.8%	32.7%

Tables 1 and 2 depict the results of the evaluation of both projects. The data show the share of the students who reached the respective percentage rating for their projects in each group.

The results of the first project were more satisfactory than we expected. After receiving comments from their classmates, the students corrected many defects in the projects and achieved a really high rating from the teacher. More than 73% of projects achieved a percentage rating corresponding to the mark "excellent". More than one third of all projects were evaluated by the teacher as exceptionally

good and exceeded her expectations. Only three students received 0% of the project – two of them have not submitted their projects at all, and the third case was a plagiarism.

As already mentioned above, the second project was considerably more challenging for the students than the first one. The results correspond to this assumption as shown in Table 2. The distribution of students in particular categories of evaluation is more even than in the case of the first project and also includes categories below 50%. Nevertheless, almost 55% of students achieved the best rating, and again almost a third of the projects were identified as exceptionally good projects by the teacher. Only one student was assigned the 0% rating, again for not submitting the project.

4.2 Survey analysis

The first questionnaire was filled in by 43 students and the second one by 44 students of which in both cases approximately 61% of girls and 39% of boys (± 0.5%).

At the beginning of the questionnaire, we asked how demanding the project was from the programming point of view. The first project was not demanding (values 1-3) for 81.4% of students, and the second one for 45.5% of students (see graph in Figure 2). While we did not expect such a high number of positive responses in the first project, their decline in the second questionnaire was predictable and corresponding to the increased difficulty of the project.

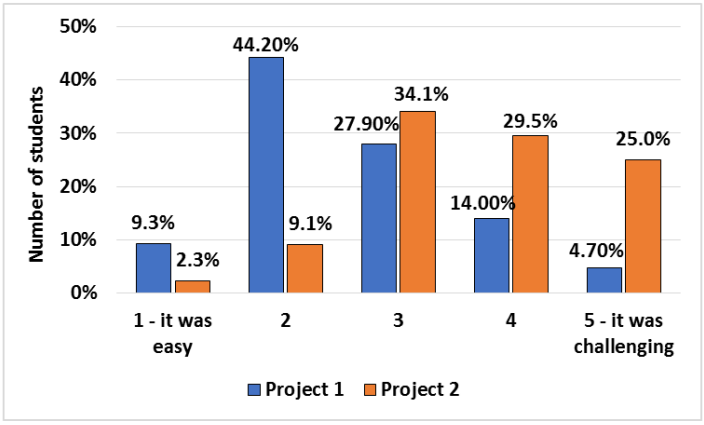


Figure 2. From the programmer point of view, was it difficult for you to develop the project? Scale from 1 (it was easy) to 5 (it was challenging)

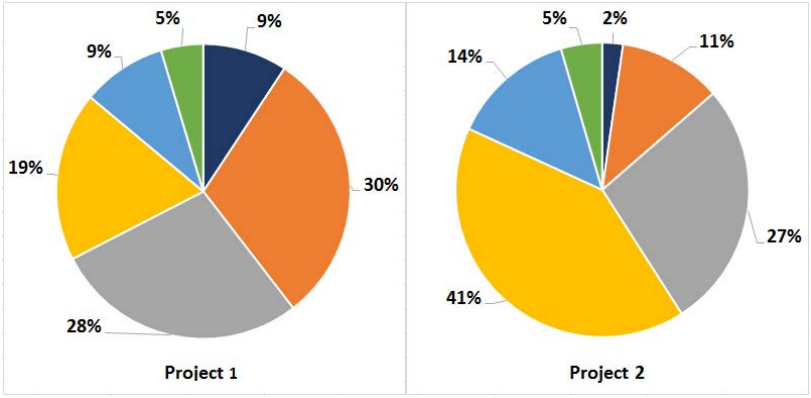


Figure 3. Was the project development time-consuming for you? No, it didn't take me long, It was adequately time demanding for me, It took me a little more time than I expected, Yes, it took me much more time than I originally expected, It was a time killer, Other

Already during developing the first project, students complained about its time demands. In the second project they even required an extended time to program it. Therefore, in the questionnaire we also examined how time-consuming both projects for students were.

As shown in Figure 3, the percentage of students who perceived the project as not very time-consuming decreased to one third in the second project compared to the first one (from 39% to 13%),

while the percentage of students who considered the project very time-consuming almost doubled in the second questionnaire compared to the first one (from 28% to 55%). More detailed information about the time spent on project programming is shown in the graphs in Figure 4.

The teacher estimated that the students should program the first project within 3 hours and the second one in 4-8 hours. In estimating it, she took into account a number of similar tasks students have programmed already as well as skills acquired in programming lessons at school and by programming the homework. If it took more time for the student, it indicates she should spend more time learning to program. According to the first questionnaire, up to 77% of students managed to program the first project in less than 3 hours. For the second questionnaire, this figure fell to only 18%, but 73% of students managed to program the project below the estimated upper limit – 8 hours (Figure 4).

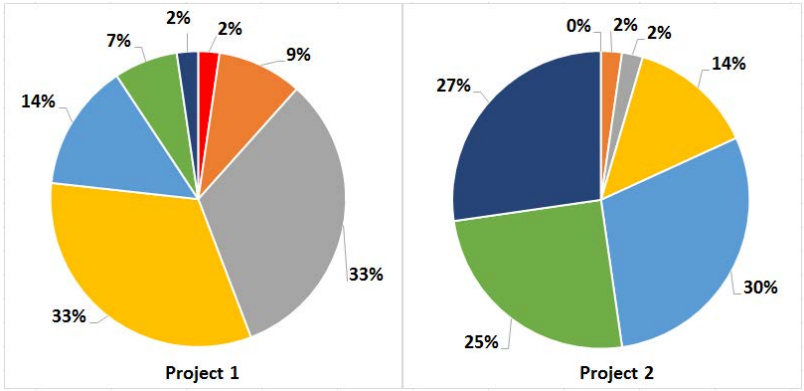


Figure 4. How long did it take to you to program your project? Up to 30 min, Over 30 min but up to 1 hour, Over 1 hour but up to 2 hours, Over 2 hours but up to 3 hours, Over 3 hours but up to 4 hours, Over 4 hours but less than 8 hours, 8 hours or more

Before developing the first project the students reviewed just short programs consisting of only a few lines of code (small reviews). Therefore, we were interested in their opinion on the complexity of commenting on the project. The question was also included in the questionnaire for the second project. Although the distribution of responses in the second questionnaire was not as explicit as in the first one (Figure 5), the share of students who did not consider the commenting on the classmate's project as challenging (values 1-3) was very high and roughly the same in both cases (90.8% and 88.6%).

The teacher also confirmed that in both projects, the students learned to understand somebody else's code, they were capable to find errors in it and comment on it. They even managed to find a mistake that the teacher did not notice when checking the first version of the program. In the first project, most of the students detected almost all errors in the classmate's project. In addition, there were many of them who have found all errors. In the second project, due to its higher difficulty, some students were unable to advice the project author how to fix the bugs in the program, but many of them managed it, which the teacher appreciated when giving feedback to this activity.

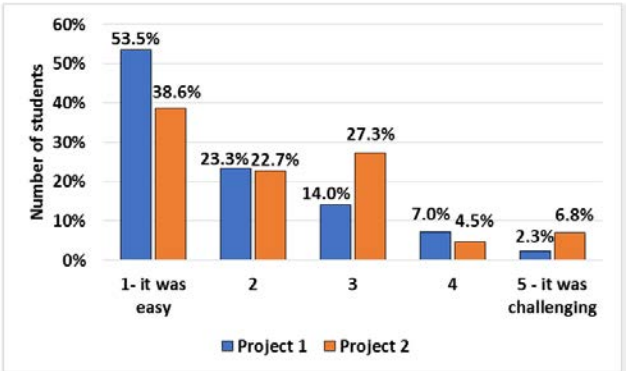


Figure 5. Was it difficult to comment on the classmate's project? Scale from 1 (it was easy) to 5 (it was challenging)

Since the students had certain experience of code-reviewing in the period of commenting on the second project we asked them in the second questionnaire to compare the difficulty of commenting on the first and second projects (Figure 6). For 52% of students, it was easier to comment on the second project, either because they already commented on the project (34%) or because they rated themselves to be better programmers who better understand the code than in the first project (18%). The proportion of students, who perceived the commenting on second project more difficult than the first commenting, was 18%. They mostly rationalized it by the second project being harder. 12% of students added their own response – with about half of them stating it was easier to comment on this project and it was more difficult for the other half. Overall, commenting on the harder project was easier for nearly 60% of students and it was more difficult for less than a quarter of students.

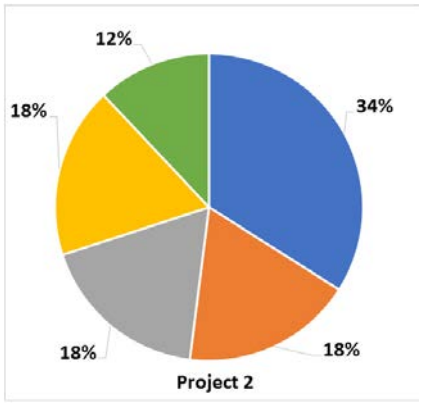


Figure 6. How difficult was it to comment on this project compared to the commenting on the previous project? It was easier just because we already commented on the project, It was easier because I am better in programming and better understand the code now than in the first project, It was as hard as in the previous project, It was harder because the project was harder, Other

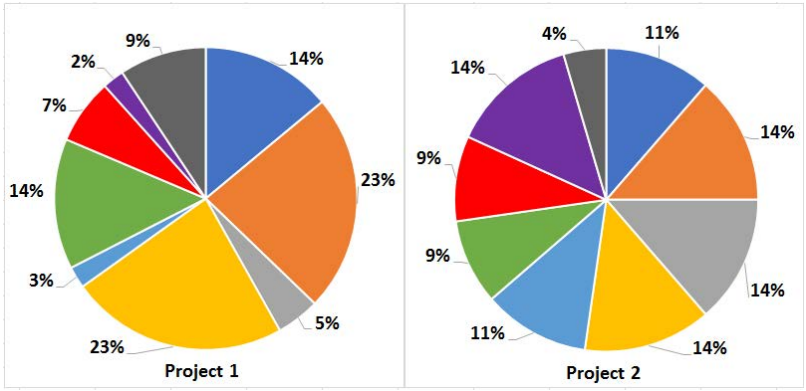


Figure 7. Did the comments from your classmate help you when finishing the project? Yes, I got good comments that helped me a lot. Yes, I received good comments that helped me, and I was delighted to be praised for the project. I received good comments but did not know how to integrate them into the project. Some comments helped me, some did not. My project worked, my classmate found no mistakes, so I did not get any comments that I could work with. The reviewer did not find any mistakes, but I was pleased that he praised my project. No, they did not help me because I did not understand them and did not know how to integrate them into the project. No, I have received nonsense comments that have not helped me at all. Other

We were also interested in whether the students considered the classmate's comments helpful when completing the project (Figure 7). In the first questionnaire, more than half of students declared positive experience of classmate's comments and only 9% of students have not considered the received comments helpful. In the second questionnaire, students' responses were almost completely evenly distributed into all categories of answers. Apparently, due to the demandingness of the project, the percentage of students who failed to clearly and meaningfully comment on their classmate's project rose to 23%. However, the amount of comments rated positively by students (39%) was still higher than those rated negatively.

Regarding the benefit of comments received from their classmates (Figure 8.), the students saw it especially in helping them correct errors they had not noticed or were unable to correct by themselves before the first submission of the project, or otherwise helped them (58% for the first and 41% for the second project). Also, the responses to this question seem to have been affected by the higher difficulty of the second project – the share of students, who stated that the comments did not help them, increased from 23% in the first questionnaire to 41% in the second one.

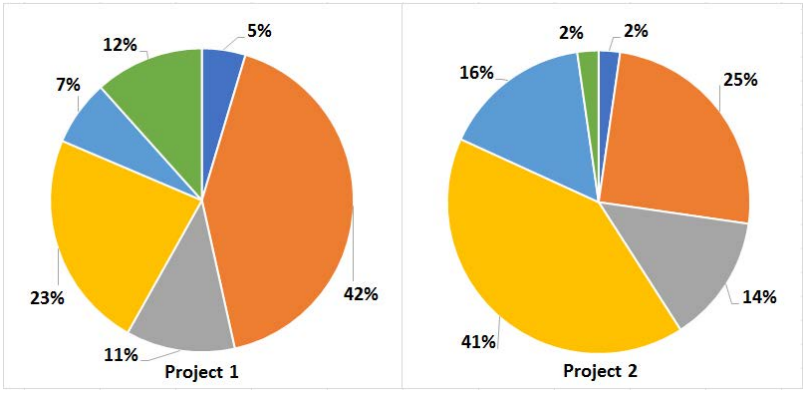


Figure 8. What were the benefits of the comments you received from the classmate? They helped me better understand the project assignment. They helped me to fix errors I did not notice before the first submission of the project. They helped me to fix errors I failed to fix before the first submission of the project. They did not help me at all. I have not received any relevant comments. Other

Since we believed that students could also benefit from commenting on the projects of the others, we were curious whether they perceived the process of commenting as beneficial to them too (Figure 9.). In both questionnaires, the most frequent answer (37.2% and 27.3%) was that they became aware of various errors that might occur in a program. Further, the students often indicated that they learned how the classmate understood the assignment, gained experience with code reviewing, and learned to give a constructive critique. For both questionnaires, roughly the same percentage of students (16.9% and 15.9%) said that they did not gain any profit from commenting on the classmate's project.

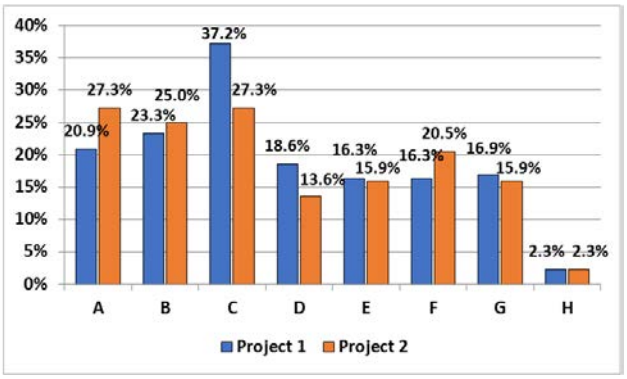


Figure 9. Do you think you benefited from commenting on other project? A: I gained experience in testing programming projects, B: I found out how my classmate understood the project, C: I became aware of various errors that might occur in a program, D: While reviewing a foreign project, I realized the shortcomings in my own project, E: I trained my verbal skills, F: I learned to give a constructive critique, G: I have not benefited from commenting at all, H: Other

At the end of the questionnaire, we investigated to what extent the students were satisfied with how they elaborated the project. As shown in Figure 10, with the first project, students were clearly satisfied, while in the second case their responses were more evenly divided into several categories. It is possible that comments from classmates helped them to better understand the shortcomings of their project.

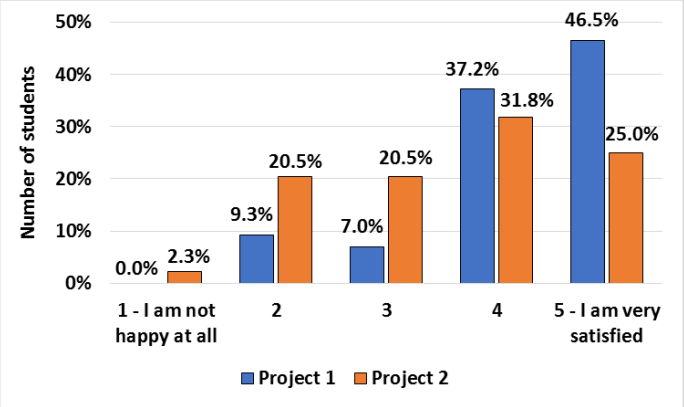


Figure 10. How satisfied are you with your completed project? Scale from 1 (I'm not happy at all) to 5 (I'm very satisfied).

5 CONCLUSIONS

In our study we tried to find out whether code review techniques can be used with high school students and whether such activities have an impact on their learning.

We examined the results of students' work on two programming projects, which included code-reviewing. Although the projects were of different difficulty, the results of their analysis showed that in both cases many students were able to comment on their classmates' program, find errors and advise how to correct them. On the other hand, thanks to the comments received from their classmates, many students have corrected errors in their programs and have achieved better assessment from the teacher.

To find out the opinions of the students on the use of this activity in the classroom teaching, we conducted a questionnaire survey. Its outcomes showed that students generally did not consider commenting on a classmate's project to be challenging and appreciated the contribution of comments that helped them to identify errors they had not noticed before, or to correct errors they were unable to correct by themselves.

From personal discussions with students, we know that from the beginning they considered the code review to be a time-consuming activity perceived as an extra work, but later realized its benefits and usefulness. Some of them stated that commenting on the other student's program also made them think about the programming issues they did not encounter in their own project.

In the teacher's opinion, the students learned to read and understand someone else's program code and to look for and correct errors in it. As she found out during the subsequent lessons, it helped them to be more independent and self-reliant in coding their own programs because they were able to better detect and correct errors and also prevent them.

Based on the results of our study, we can recommend the use of code review as an educational activity at high school. Students learn to read and understand someone else's code, to detect errors in the program, and perceive different approaches to solving the same problem.

ACKNOWLEDGEMENTS

This work was supported from Slovak national project VEGA 1/0797/18.

REFERENCES

[1] H. Purchase and J. Hamer, "Perspectives on peer-review: eight years of Aropä". *Assessment & Evaluation in Higher Education*, vol. 43, no. 3, pp.473-487, 2018.

[2] V. Dropčová and Z. Kubincová,: "Team-based projects and peer assessment. it works!", In *International Conference on Interactive Collaborative Learning*, pp. 112-127, Springer, 2016

[3] S. C.Tseng and C.C.Tsai, "On-line peer assessment and the role of the peer feedback: A study of high school computer course", *Computers & Education*, vol. 49, no. 4, pp.1161-1174, 2007.

[4] K. J. Topping, “Peer assessment between students in colleges and universities”, *Review of Educational Research*, 68, pp. 249-276, 1998.

[5] E. Z. F. Liu, S. S. J. Lin, C.H. Chiu, S. M. Yuan, “Web-based peer review: the learner as both adapter and reviewer”. *IEEE Transactions on Education*, vol. 44, no. 3, pp. 246–251, 2001

[6] Z. Kubincová, M. Homola and V. Bejdová, “Motivational effect of peer review in blog-based activities”, In: *ICWL 2013*. Kenting, Taiwan. Springer, LNCS, vol. 8167, 2013.

[7] R.K. Ladyshewsky, P. Gardner, “Peer assisted learning and blogging: A strategy to promote reflective practice during clinical fieldwork”, *Australasian Journal of Educational Technology,* vol. 24, no. 3, 2008.

[8] T. Gilb and D. Graham, *Software inspection*. Reading: Addison-Wesley Longman Publishing Co, 1993.

[9] Boehm, B. W., *Software engineering economics*. Prentice-hall Englewood Cliffs (NJ), 1981.

[10] K. E. Wiegers, *Peer reviews in software: A practical guide*. Addison-Wesley Boston, 2002.

[11] N. Falchikov, J. Goldfinch, “Student peer assessment in higher education: a meta-analysis comparing peer and teacher marks”, *Review of Educational Research*, 70, pp. 287-322, 2000.

[12] Z. Kubincová and M. Homola, “Code review in computer science courses: Take one”, In *International Conference on Web-Based Learning*, pp. 125–135, Springer, 2017.

[13] Z. Kubincová, and I. Csicsolová, “Code Review in High School Programming”, In *17th International Conference on Information Technology Based Higher Education and Training* (ITHET), pp. 1-4, IEEE, 2018.