Use of Automatic Code Assessment Tools in the Programming Teaching Process

Marílio Cardoso[1]
Department of Informatics, ISEP, Polytechnic of Porto, Portugal
joc@isep.ipp.pt

António Vieira de Castro
Department of Informatics, ISEP, Polytechnic of Porto, Portugal
avc@isep.ipp.pt

Álvaro Rocha
Department of Informatics, Faculty of Science and Technology, University of Coimbra, Portugal
amrrocha@gmail.com

Emanuel Silva
Department of Informatics, ISEP, Polytechnic of Porto, Portugal
ecs@isep.ipp.pt

Jorge Mendonça
Department of Mathematics, ISEP, Polytechnic of Porto, Portugal
jpm@isep.ipp.pt

Abstract

The teaching of programming process is essential to prepare students for the development of computer applications and software solutions. During the last decade, a variety of tools facilitating automatic validation of programming code have been developed. In this context, authors start to analyze and studying some tools with this potential and a possible use with pedagogical purposes. For the last three years a study has been carried out related with the implementation of VPL (Virtual Programming Lab) a plug-in developed specifically for Moodle (Modular Object-Oriented Dynamic Learning Environment) on a Java-based programming discipline during the Informatics Engineering degree of the Informatics Engineering Department (DEI) from the School of Engineering of Polytechnic Institute of Porto (ISEP/P.PORTO). This paper will present how VPL was introduced and some results of this experiment before the implementation in the learning process of another tool (Mooshak) as a real-time automatic code evaluation. These tools allow to edit and execute programs, in a large range of languages, and enables automatic assessment and prompt feedback.

2012 ACM Subject Classification Software and its engineering; Applied computing → Interactive learning environments

Keywords and phrases Teaching programming, APROG, Moodle, VPL, Mooshak, Automatic assessment

Digital Object Identifier 10.4230/OASIcs.ICPEC.2020.4

1 Introduction and context

Programming is, nowadays an important skill even for those who do not work or intend not to work in information and communication technologies area (ICT).

Besides it could be relevant to personal development, it can also help in professional activity, allowing to better understand the context of technology, automate and optimize tasks.

[1] Corresponding author

First International Computer Programming Education Conference (ICPEC 2020).
Editors: Ricardo Queirós, Filipe Portela, Mário Pinto, and Alberto Simões; Article No. 4; pp. 4:1–4:10
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 Programming skills in modern society

Learning programming in early ages increases logical reasoning, improves cognitive abilities and human interactivity and transforms perception to establish logical connections giving also self-confidence and problem-solving skills [1].

On the other side, computer programming teaching is a complex and challenging task for any teacher, mainly for those who have beginner students, since this activity is very different from teaching any other subject, so the approach will be different.

1.2 The traditional programming Learning process

On the "write code" process, the programmer must "translate" the algorithm into a programming language. A programming language is a formal language with a specific set of well-defined instructions and syntax rules, which enable software development. There is a huge set of programming languages, each one with specific syntax and application fields.

According Cardoso et. al. [3], the TIOBE[2] index, that is a Programming Community index, is a good indicator of the popularity of programming languages and considering that it is one of the most cited related indexes, their information provides a global overview about the programming languages usage. Their ratings are monthly updated and are based on the number of skilled engineers world-wide, courses and third-party vendors.

Cardoso et al. also refer PYPL (PopularitY of Programming Language)[3], another index about programming languages that is based on Google and the Redmonk[4] that is a ranking of programming languages obtained from information about GitHub code lines and "StackOverflow"[5] language tags.

Nowadays, Java, Python and C# appear in the top 5 of recent rankings of these indices.

In Informatics Engineering degree of the Informatics Engineering Department of School of Engineering (ISEP), Polytechnic Institute of Porto (P. PORTO), Java was the chosen language to introduce novice students to programming.

APROG (Algorithms and Programming) is the first course unit that directly deals with programming skills, where students start coding with Java using NetBeans, an Integrated Development Environment (IDE).

1.3 The case of the APROG course unit

Figure 1 presents APROG organization with three different types of classes: theoretical (T), theoretical-practical (TP) and practical-laboratorial (PL).

This course unit takes place during the first semester of the first year and has usually about three hundred students and several teachers (usually between 6 and 10).

We must grant that lessons are similar inside each class and with its important to ensure that pedagogical goals will be the same, independently of the teachers allocated to each class. For this, a good planning and organization is needed as well as strong and clear coordination.

[2] https://www.tiobe.com/tiobe-index

[3] http://pypl.github.io/PYPL.html

[4] https://redmonk.com/

[5] https://pt.stackoverflow.com/

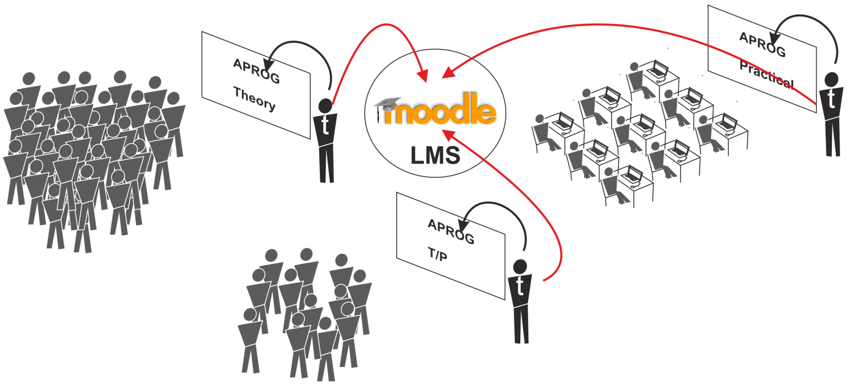


Figure 1 Types of APROG classes.

On the Table 1 are presented the different weekly type of classes as well as the number of hours allocated to it.

Table 1 Weekly class types.

Class type	Number of classes by week	Hours by class	Total week hours
T	1	1	1
TP	1	1	1
PL	2	2	4

In the last three academic years (2017–2018, 2018–2019 and 2019–2020), in APROG, were allocated respectively 10, 6 and 8 teachers for a population of 318 students in 2017–2018, 307 in 2018–2019 and 304 in 2019–2020.

The teacher responsible for APROG has changed on 2019–2020 and his responsibility is to teach theoretical classes. Other 3 teachers were involved in theoretical-practical classes and they also teaches practical-laboratorial classes. The distribution of teachers by type of class is shown in Table 2.

Table 2 Teachers by class type.

Class type	T	TP	PL
2017/18	1	3	9
2018/19	1	3	5
2019/20	1	3	7

In the practical-laboratorial classes a methodology named eduScrum [4] was applied where students were organized in work teams. EduScrum methodology is a variation of Scrum [12] but applied with pedagogical purposes.

Scrum is a method that has been used to manage the development of software and many other complex products, mainly by software companies to promote teamwork and increase productivity and creativity on the team members. Its use is being tested in many other areas, and education is one of them [6].

In Portugal, as well as in the rest of Europe, one of the most popular and widespread Learning Management System (LMS) is Moodle, that is an open source solution delivered under General Public Licenses (GNU). Since 2006 ISEP also adopted Moodle and it is widely used to place organized educational contents. Moodle allows the submission of works that has been assigned to students and promote communication between teachers and students and do students surveys.

In APROG course unit, new content is weekly organized and in each week are included in Moodle a set of theoretical content and a set of practical exercises that students must solve.

In each class teacher analyzes the resolutions provided by the students and give them a feedback. Each class has about twenty students and each worksheet have six or more exercises. This activity represents a big challenge for APROG teachers, because the feedback must be provided weekly, resulting in a great overhead of tasks. With such a big set of tasks its almost impossible to evaluate deeply all of them. The existing solution is to evaluate by sampling, and this means that the feedback is incomplete and with some delay. This prevent students to continue developing their works and improve faster their programming skills.

The lack of feedback and mentoring from a teacher to students is one of the most important questions related with learning about programming [11], that is related and may contribute to students' lack of motivation and commitment.

We identified this as a problem and we wanted to contribute to solve it urgently since teachers do not have enough time to keep up with students, and also, students can practice coding without having to wait a few hours or even days for the teacher validation of their code.

Our first impulse was to propose a reduction of the number of students per class as well as the number of exercises that they must solve but, due the importance of practice for develop programming skills, teachers agreed to not reduce it.

So, we conclude that something should be done to help, and we began to study and identify new ways to reduce the time spent by teachers in verifying and evaluating assignments and helping students to improve their programming skills more quickly.

In this demand we identified a myriad of potential tools for our purpose. However, we defined a set of requirements, according to the conditions we had, the needs and characteristics that we thought desirable for the tool to choose. These requirements were: to be a free tool, integrated to Moodle, easy to use, suitable for teaching programming, allowing code upload, with plagiarism detection capabilities, with security concerns and supporting Java. With these characteristics we choose Virtual Programming Lab (VPL) to implement a study and realize a preliminary pedagogical experience.

2 The VPL case in APROG

As mentioned, to analyze the use of automatic code validation tools and the study of their potential, a pedagogical perspective started in a study that was based on the use of VPL.

2.1 The VPL plugin

VPL is a Moodle plugin developed by the Department of Computer Science and Systems from University of Las Palmas of Gran Canaria, Spain and it is an open source solution under the GNU/GPL license [11].

This is a tool that allows to manage programming assignments on a large set of programming languages. VPL can identify the language in an automatic way.

The VPL architecture has the following three main specifications: [2]

- a Moodle module with specific features such as: submission management, assessment support and anti-plagiarism;
- a browser-based code editor, which allows coding and execute programs;
- a Linux server (jail server) which hosts the environment where the students' assignment will be executed and evaluated in a remote and secure way.

With VPL its possible to manage computing programming assignments and provide a mechanism for automatic grading. Other functionalities allow to edit, compile, run, debug and evaluate code, with prompt feedback and at same time stores historic results related with compilation and implementation of the assignment and track the student's submissions.

2.2 Process Implementation

Several steps were panned to implement the process, as well a list of conditions that must be considered. One of the most important was that the experience must not disrupt the existing pedagogical process, that is, it should not disturb the normal course of classes and at same time, should assure impartiality on the evaluation to all students.

It was defined that VPL should be available in the institutional LMS (Moodle), so it installation was carried out by the responsible technician for the internal Moodle management.

All the steps were carefully planned to grant a soft, robust and progressive implementation of this study. On Figure 2 we present the main phases on the process.

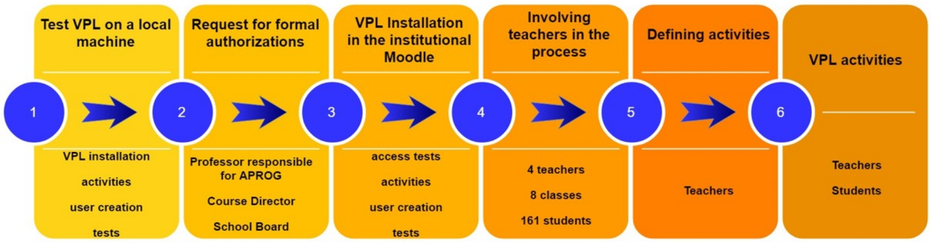


Figure 2 Planning process phases.

From the set of exercises on each week are defined some activities and created adequate assignments. After identifying the exercises, it was necessary to prepare their statements and test cases, implement and test with a dummy user with student role.

Finally, VPL will be available for all students and assignments will be possible.

2.3 VPL activities

Were selected six exercises to use VPL, and the necessary adaptations were made to the statement, in order to reduce, as much as possible, any ambiguities and referring some particularities to be considered in the submission. Examples of the input data format and the expected results have also been added, in order to smooth the use and reduce the probability of error.

The students prepared the resolutions of the exercises in the IDE and, for the mentioned six exercises, proceeded to their submission in the VPL. In each submission, the system gives feedback, allowing the student to see if the result obtained was the expected one and, if not, to identify the tests in which he/she failed, allowing him/her to redo the resolution and resubmit.

2.4 Some results related with VPL

VPL was used in the academic years 2017–2018, 2018–2019 and 2019–2020.

However, the presented results refer only to the academic year 2018–2019. This is due to the fact that, in that year, the number of students involved was significantly higher than in the previous year, the teachers had more experience using VPL and the (opinion) survey carried out was answered by a larger number of students becoming more complete and consistent. Data of the academic year 2017-2018 were processed by Excel and SPSS (version 25).

With this study, the matrices of Spearman's nonparametric correlations and Cronbach's alpha were determined, this parameter being an indicator of the internal consistency of the survey. On the survey, to be considered reliable, Cronbach's alpha must be greater than 0.7 [10]. In the 2017–2018 survey, a Cronbach's alpha of 0.614 was obtained, thus showing a poor reliability, so we do not report these results. In the year 2018–2019 the survey was revised and adjusted so that a Cronbach's alpha of 0.822 was obtained, which reveals the reliability of the study and the results presented.

In the academic year 2019–2020, VPL was used only as a complement to the Mooshak, and no opinions were collected from students about its use.

Thus, regarding the survey carried out in 2018–2019, we present results on some issues that seem to be more important.

In the surveys, a five-level Likert scale was used (Figure 3).

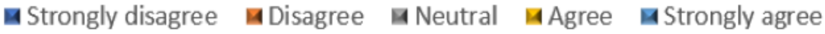
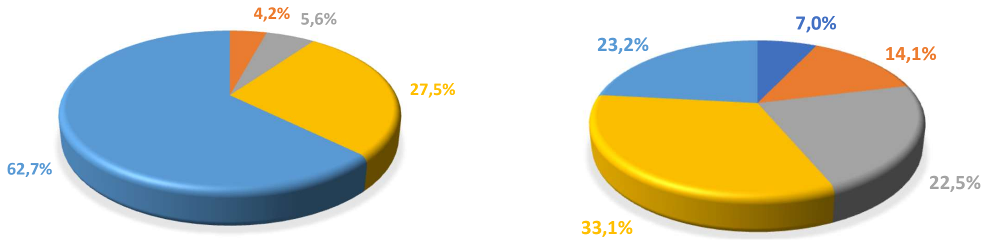


Figure 3 Likert scale.

One of the questions intended to collect the students' opinion about the use of Distance Learning platforms such as, for example, Moodle. To this question, more than 90% of respondents answered that they totally or partially agree, with only 4.2% partially disagreeing, as can be seen in Figure 4a. It was also asked whether VPL was useful in the learning process. Most students (56.3%) agree (totally or partially) with the statement, with 21.1% of respondents saying that they disagree partially (14.1%) or totally (7.0%) (Figure 4b).



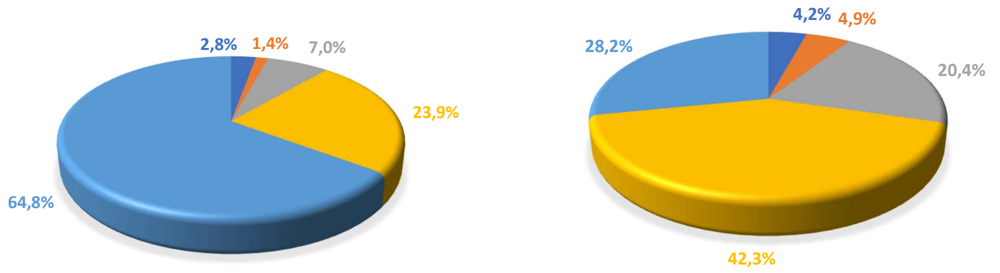
(a) Distance learning platforms (such as Moodle) are an added value.

(b) Use VPL for solving programming exercises was helpful in my learning process.

Figure 4 Survey responses about learning.

In the graphs presented in Figure 4, it is possible to observe that there is a great interest from students and a willingness to use technologies in the educational process. It is also possible to infer that most students believe that VPL was useful in their learning process.

With VPL it is possible to obtain an evaluation and, in case it does not correspond to the maximum value, the system presents the results of the failed tests, assisting the student to understand what needs to be corrected. As the automatic classification associated with the possibility of resubmission of works is one of the functionalities of VPL, it was intended to know the students' opinion about its usefulness, (Figure 5a), with 88.7% of the respondents saying that they totally or partially agree. It was also intended to gather the opinion on the added value of VPL for the teaching-learning process of programming. Only 9.1% of the students answered this question negatively, with 70.4% of the students totally or partially agreeing with the statement (Figure 5b).



(a) Be able to send assignments several times and get an automatic note is a very useful feature.

(b) VPL is an added value to the teaching-learning process of programming.

Figure 5 Survey responses about VPL usage.

From the analysis of the graphs presented in Figure 5, regarding the teaching-learning process, an extremely positive global appreciation by the students is evident, with a slight degree of disagreement regarding the statements of questions "VPL is an asset for the teaching-learning programming process" and "The use of VPL in solving exercises was an aid to my learning process". These are very rewarding results for the effort and dedication dedicated to the study and motivators for the future use of VPL in teaching programming.

3 Use of Mooshak in APROG

In addition to the study carried out with the VPL, we intend to test the potential of a new tool called Mooshak with which we have already had some contact and previous experience with programming contests.

3.1 The Mooshak application

Mooshak is a client–server application to fully manage and run programming contests [7]. It is a Web-based application and all of its functionalities are accessible by a Web-browser, and independent of Operating Systems. It is an open source system originally designed to manage online programming contests. The development of Mooshak was based on the rules of the ACM International Collegiate Programming Contest (ACM-ICPC) [8]. These contests are oriented to teams of students from higher education institutions from around the world [5]. Mooshak has been used in various programming contests such as SWERC (South Western Europe Regional ACM Programming Contest), MIUP (Inter-University Programming Marathon), TIUP (Inter-University Programming Tournament) or ToPAS (Tournament Programming Contest for Secondary Students). In the meantime, with the updates introduced, it is now possible to manage competitions from other fields and with different rules, namely, the Portuguese section of the International Olympiad in Informatics (IOI) [8].

A contest is a set of exercises students must solve using a programming language in a time window. Students submit source code for problem solving using a browser. Mooshak receives the source code and automatically compiles it, executes it using a set of pre-prepared tests and provides immediate feedback. The solution submitted to solve the problem is accepted if it satisfies all secret test cases prepared in advance for the exercise concerned. Otherwise, an error message will be displayed based on compilation / execution state.

Given the enhanced capabilities of automated code evaluation, Mooshak has been increasingly used as a pedagogical tool in teaching programming [9].

3.2 Mooshak – past experience

Since 2001, some programming competitions have been held among ISEP Computer Engineering students. Students loved the experience and were motivated to pursue other competitions. The best classifieds could participate in inter-university (MIUP) and international (SWERC) competitions.

Figure 6 illustrates a programming contest held at DEI-ISEP in 2006 and mediated with Mooshak.

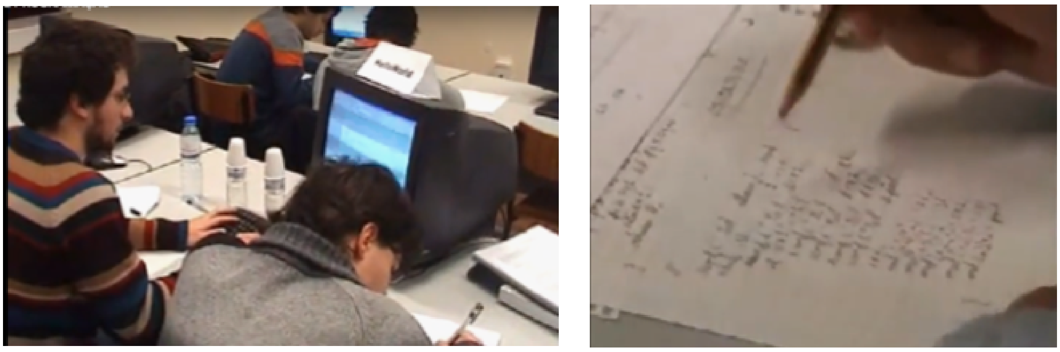


Figure 6 Programming contest on 2006 at ISEP.

The students consider that their participation in programming contests is an enriching personal experience and a relevant curriculum value. Given this environment of motivation and enthusiasm, it was considered interesting to replicate this experience in the classroom and try to foster this feeling for all students[6].

3.3 Contest in the classroom

Due to the background of the programming contests, it was intended to bring this motivation into the classroom and promote a kind of gamification in which students are valued for their performance in solving exercises and develop healthy competition among colleagues.

In addition, students have heterogeneous knowledge and distinct work speeds. These factors lead to the resolution of the problems proposed in class not being achieved at the same time by the students. An assessment tool gives students immediate feedback on their solutions.

Faced with large classes, the teacher is unable to give feedback to all students immediately after solving their problems. Waiting for teacher availability for resolution feedback can lead to a drop in student concentration and productivity. To minimize this concern, Mooshak can be used as the first line tool capable of giving immediate feedback to the student in solving the proposed exercises. When the teacher is available to the student, he or she can provide personalized feedback, not just in a particular exercise, but on all exercises already solved and validated in the first phase by Mooshak. Resolutions submitted and accepted by Mooshak are then reviewed by teachers to determine whether the answer is appropriate or, despite being validated by Mooshak has an inappropriate approach to the taught content. In this case the student is oriented to try a new resolution approach and the accepted submission may be revoked.

[6] Contest 2006 video available in `https://www.youtube.com/watch?v=xzfpmVey8BU`.

Each problem has a set of secret tests with an associated score. The submitted solution is graded with the sum of the successful test score. The submission score only reflects successful tests and not successful in all tests. The idea is that while the solution may only be a partial solution and not solve all scenarios, it is intended to enhance student effort and encourage them to look for a better solution for a better score.

The type of problems proposed try to follow the style of exercises of the international programming competitions (ACM-ICPC). A task is presented for the student to solve. One of the main concerns is to educate the student to respect what is requested and to be rigorous in their response. It is essential to respect strictly the format required in the job description. Any mistake, even slight, is enough for a solution not to be accepted. In all exercises, examples of input and their expected output are provided. This way, the student always has at least one example to test their program as intended.

The syllabus of the course is divided into three parts. For each part a contest is created with a set of exercises appropriate to that content and students are challenged to participate. The duration of the contest corresponds to the duration of the related program content.

In addition, Mooshak has some interesting features, namely, it provides a set of statistics that allow the student to track their performance and progress over time, as well as compare with peers through a ranking.

3.4 Results with Mooshak

At the end of the APROG course, 195 of the 304 students enrolled answered an anonymous survey to rate some questions on a scale of 1 to 5, where 1 means "strongly disagree" and 5 means "strongly agree". Table 3 summarizes the answers to the questions about using Mooshak.

It is observed that in general, the students positively appreciated the use of Mooshak. On the first question, 78% of students agreed, or strongly agreed, that automatic feedback was an asset in solving exercises. Also relevant is that 83% of students feel that using Mooshak motivated them to solve problems.

Table 3 Results of survey about using Mooshak.

	1	2	3	4	5
The use of Mooshak was very important as an aid in the feedback of the exercises	2%	5%	15%	41%	37%
The availability of Mooshak in class activities motivated you to be more active	2%	6%	10%	40%	43%

1 – Strongly Disagree; 2 – Disagree; 3 – Neutral; 4 – Agree; 5 – Strongly Agree

4 Conclusions and future work

Based on student responses to surveys, it is considered that the use of automatic assessment tools, and particularly Mooshak, has a considerable impact on students' motivation and active participation and can therefore significantly enhance the aid to teaching and learning programming.

One of DEI's strategic objectives is to continue new programming contest events.

In this sense, it is intended to consistently use a tool designed for programming contests to promote student training and increase their dexterity in solving competition style problems.

The experiments carried out with the VPL also demonstrated its great potential for use, with good acceptance by the students.

As VPL and Mooshak are two tools designed for different purposes, their use in APROG proved to be effective in a complementary perspective, since VPL was used after Mooshak and with more refined tests.

Nowadays, with a global job market, there is a growing trend in the use of automatic code validation tools, for the selection and recruitment of programmers. Thus, its use in academic and teaching contexts will represent an added value for students to enter the job market.

We believe that this study is opening doors to a new paradigm of programming teaching that will provide to involve student's motivation and new feedback mechanisms that will pass them the grant of a good performance giving them freedom to keep going.

References

1 Marina Umaschi Bers, Louise Flannery, Elizabeth R. Kazakoff, and Amanda Sullivan. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72:145–157, 2014. doi:10.1016/j.compedu.2013.10.020.

2 Julio C Caiza and Jose M Del Alamo. Programming assignments automatic grading: review of tools and implementations. In *7th International Technology, Education and Development Conference (INTED2013)*, page 5691, 2013.

3 J.Marílio Cardoso, António Vieira de Castro, Rosa Barroso, Álvaro Rocha, and Rui Marques. Introducing vpl on a programming learning process. In *EDULEARN18 Proceedings*, 10th International Conference on Education and New Learning Technologies, pages 8499–8508. IATED, 2–4 july 2018. doi:10.21125/edulearn.2018.1981.

4 Eduarda Pinto Ferreira and Ângelo Martins. Eduscrum—the empowerment of students in engineering education? In *Proceedings of the The 12th International CDIO Conference*, page 596, Turku University of Applied Sciences, Turku, Finland, 2016.

5 ICPC. The icpc international collegiate programming contest. Accessed on 20.01.2020. URL: https://icpc.baylor.edu/.

6 Craig Larman. *Agile and iterative development: a manager's guide*. Agile software development series. Addison-Wesley Professional, Boston, USA, 2004.

7 José Paulo Leal and Fernando Silva. Mooshak: A web-based multi-site programming contest system. *Software: Practice and Experience*, 33(6):567–581, 2003.

8 José Paulo Leal and Fernando Silva. Using mooshak as a competitive learning tool. In *The 2008 Competitive Learning Symposium*, 2008.

9 Xiangfeng Luo, Marc Spaniol, Lizhe Wang, Qing Li, Wolfgang Nejdl, and Wu Zhang. *Advances in Web-Based Learning-ICWL 2010: 9th International Conference, Shanghai, China, December 8–10, 2010, Proceedings*, volume 6483. Springer, 2010.

10 João Maroco and Teresa Garcia-Marques. Qual a fiabilidade do alfa de cronbach? questões antigas e soluções modernas? *Laboratório de psicologia*, pages 65–90, 2006.

11 Juan Carlos Rodríguez-del Pino, Enrique Rubio Royo, and Zenón Hernández Figueroa. A virtual programming lab for moodle with automatic assessment and anti-plagiarism features. In *Proceedings of the 2012 International Conference on e-Learning, e-Business, Enterprise In-formation System*, 2012.

12 Jeff Sutherland. *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business, 2014.