

Criteria for selecting open web-oriented technologies for teaching the basics of programming to future software engineers

Tetiana A. Vakaliuk^{1,2,3}, Oleg M. Spirin^{4,2} and Valerii V. Kontsedailo⁵

¹Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine

²Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

³Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

⁴University of Educational Management, 52A Sichovykh Striltsiv Str., Kyiv, 04053, Ukraine

⁵Inner Circle, Nieuwendijk 40, 1012 MB Amsterdam, Netherlands

Abstract. The article presents the criteria for selecting open web-oriented technologies for teaching the basics of programming to future software engineers. An analysis of the available open web-oriented technologies for teaching the basics of programming to future software engineers made it possible to divide them into compilers, automated systems for checking programming tasks, mind maps, and massive open online courses. Criteria and corresponding indicators are defined for each of them: for web-oriented compilers and mind maps, design and functional criteria are allocated, for web-oriented automated systems for checking programming tasks, design, information-didactic, and communication criteria are allocated, for massive open online courses – informational didactic and functional. Comparative tables are given for all the listed web-oriented technologies for teaching the basics of programming to future software engineers according to certain criteria and indicators.

Keywords: programming, basics of programming, web-oriented teaching technologies, selection criteria

1. Introduction

In the context of European integration and the development of the higher education system, the problem of developing the creative personality of a future specialist in the professional training, including software engineers, is of particular importance. At the same time, the goal of this type of training for future software engineers should be subject to the general tasks of training, educating, and developing the personality of a future specialist, primarily due to the transition to a new information society.

As noted by Kremen [8], digitalisation of the educational process is the calling of our time, since education beyond electronic communication, information retrieval and processing based

✉ tetianavakaliuk@gmail.com (T. A. Vakaliuk); oleg.spirin@gmail.com (O. M. Spirin); valerakontsedailo@gmail.com (V. V. Kontsedailo)

🌐 <https://sites.google.com/view/neota/profile-vakaliuk-t> (T. A. Vakaliuk);

<http://umo.edu.ua/en/university/leadership/spirin-olegh-mikhajlovich> (O. M. Spirin);

<https://www.linkedin.com/in/kontsedailo/> (V. V. Kontsedailo)

🆔 0000-0001-6825-4697 (T. A. Vakaliuk); 0000-0002-9594-6602 (O. M. Spirin); 0000-0003-2240-7891 (V. V. Kontsedailo)



© Copyright for this paper by its authors, published by Academy of Cognitive and Natural Sciences (ACNS). This is an Open Access article distributed under the terms of the Creative Commons License Attribution 4.0 International (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

on ICT and network technologies is becoming an anachronism. First and foremost, there is a need to develop high-quality electronic educational resources and create special information technology educational platforms, which will become available regardless of residence (urban or rural areas) for all participants in the educational process. Meanwhile, the computer facilities of education institutions need to be upgraded and replenished with modern mobile devices, new visualisation aids, and interactive tools. The availability of mobile communication devices and tablets to the majority of the population provides the background for extensive use of cloud information environments in the educational process, developing teachers' digital competence, and enhancing their information and communication culture. [8, p. 48]

According to the curricula that future software engineers study, there are several programming languages to learn: C++, C#, PHP, JavaScript, Python, etc. At the same time, if you install all the necessary software on a computer, the student will work in different environments, and each time you need to adjust to working with a new compiler [20]. In the era of informatization of society as a whole, web-oriented compilers with support for various programming languages began to be developed to solve this problem [20].

In such conditions, each institution of higher education must take significant steps towards the use or design of such a system that would cover the possibility of testing students' knowledge quickly and, most importantly, qualitatively. When teaching the basics of programming to future software engineers, each teacher repeatedly faced the problem of checking the correctness and efficiency of the algorithm. After all, such a process is quite difficult and time-consuming, and also takes a lot of time if it is done "manually".

An important thing in the training of future software engineers is the programming language training. Among the main web-oriented technologies for teaching the basics of programming, Vakaliuk [20] singled out: compilers, automated systems for checking programming tasks, mind maps, and massive open online courses.

2. Related work

Research of scientists in assessing the quality and effectiveness of information and communication technologies (ICT) training mainly covers the issues of assessing the results of educational activities. Thus, Bykov et al. [2], Galchevska [4], Holovnya [7], Spirin et al. [16], Tkachuk et al. [17] considered the criteria and indicators for selecting ICT for educational and scientific activities.

Cloud-based tools for learning computer science are reviewed by Abdullahi et al. [1], Glazunova et al. [5], Küçük, Bayılmış and Msongaleli [9], Liao, Chen and Xiong [10], Markova et al. [11], Panchenko and Khomiak [13], Rassovytska and Striuk [14], Segrelles Quilis, Moltó and Blanquer [15], Tryus and Kachala [18], Tseng et al. [19], Zhao and Forouraghi [22].

In particular, Panchenko and Khomiak [13] explored the possibilities of using cloud technologies for data analysis. Küçük, Bayılmış and Msongaleli [9] considered the possibility of using cloud technologies in the process of training a computer systems engineer.

Abdullahi et al. [1] explored the possibilities of using cloud technologies for checking and accounting for solutions to programming problems. Markova et al. [11] considered the methodological aspects of the use of cloud-oriented tools in the training of information technology

specialists.

Segrelles Quilis, Moltó and Blanquer [15] considered the theoretical aspects of the use of cloud technologies in problem-based learning on grid computing. Glazunova et al. [5] considered the effectiveness of cloud services for implementing a programming training project. Tryus and Kachala [18] considered using cloud technologies in the management and educational process of technical universities.

However, the problems of a comprehensive assessment of web-oriented technologies for teaching the basics of programming to future software engineers require further research with clarification of the criteria and quality indicators associated with the process and result of educational activities.

The *purpose* of the article is to define the criteria and establish appropriate indicators for the selection of open web-oriented technologies for teaching the basics of programming to future software engineers.

3. Research method

To achieve this goal, the following methods were used: studying the practical experience of using web-oriented technologies for teaching the basics of programming to future software engineers at the Zhytomyr Polytechnic State University in 2019–2021, systematization and generalization to determine the selection criteria and indicators, peer review method.

Experts were involved twice. The first stage – is to determine the most significant web-oriented technologies for teaching the basics of programming to future software engineers. The experts were deans of faculties, heads and teaching staff from departments related to teaching the basics of programming (20 people).

Using the method of expert evaluation to determine the most significant web-oriented technologies for teaching the basics of programming to future software engineers is that the corresponding learning technologies are numbered in ascending or descending order of a certain feature and ranked according to this feature. In total, 13 different web-oriented technologies for teaching the basics of programming to future informatics teachers (compilers, automated systems for checking programming tasks, mind maps) were proposed for consideration by experts to rank.

A scoring system was proposed, according to which, for $N = 13$ learning technologies, the value of N is given to the most significant in use, 1 to the least significant. The results of the surveys are summarized in a table, where the technology number is indicated in the columns, and the expert number is indicated in the cells. To prevent a psychological clue that could influence the expert's choice of certain ranking order, web-oriented technologies for teaching the basics of programming are placed on the card in ascending alphabetical order.

The main parameter for assessing the significance of an indicator is its total rank S . The total ranks of indicators are calculated by the formula

$$S_j = \sum_{i=1}^m R_{i,j} \quad (1)$$

where S_j – total rank of the j -th indicator;

$j = 1, 2, 3, \dots, n$; n – count of indicators;

m – count of experts;

$R_{i,j}$ – rank of the j -th indicator determined by the i -th expert.

However, such total ranks will be objective if there is a certain level of agreement between the experts. The degree of such agreement determines the concordance correlation coefficient W [6]. Taking into account the fact that

$$d_j = S_j - 0.5m(n + 1), \quad (2)$$

$$S(d^2) = \sum_{j=1}^n d_j^2, \quad (3)$$

and the maximum value of the $S(d^2)$ is achieved if all experts perform the ranking in the same way is $S_{max}(d^2) = \frac{1}{12}m^2(n^3 - n)$, the concordance coefficient is calculated by the formula:

$$W = \frac{S(d^2)}{S_{max}(d^2)} = \frac{12S(d^2)}{m^2(n^3 - n)} \quad (4)$$

Having made calculations according to formulas (1) – (4), based on experimental data, we obtain the value of W . If the value obtained is significantly different from zero, then it can be argued that there is an objective agreement between the experts (when $W = 0$, it is considered that there is no connection between the rankings of experts, when $W = 1$, the rankings completely coincide) and the total ranks are quite objective.

Deans of faculties, heads and teaching staff from departments related to preparation of bachelors in software engineering (14 people) acted similarly to the experts to determine the most significant for the use of platforms for massive open online courses (MOOC). The experts were offered a questionnaire to determine the most significant MOOC platforms for teaching bachelors in software engineering, where the respondents were offered 10 MOOC platforms for ranking. The calculations were made similarly.

MOOC platforms were separated from other web-oriented learning technologies because the authors wanted to hear the opinion of experts on MOOCs separately.

In the second stage, another group of experts was involved in the selection of the most significant web-oriented technologies for teaching the basics of programming: compilers, automated systems for checking programming tasks, mind maps, and massive open online courses. For this purpose, the manifestation of each of the defined criteria for each of the named web-oriented technologies for teaching the basics of programming was checked.

During academic conferences, webinars, workshops, personal meetings, round tables, e-mail correspondence, etc. with the results of using web-oriented technologies for teaching the basics of programming future software engineers, a significant number of deans of faculties, heads and teaching staff from departments related to training of future software engineers and teaching the basics of programming (more than 50 people) were involved.

However, information to check the manifestation of each of the named criteria for each of the selected web-oriented technologies for teaching the basics of programming to future software engineers was taken from a different number of respondents:

- 1) for compilers, both criteria were evaluated by 9 people;

- 2) for automated systems for checking programming tasks, all criteria were evaluated by 12 people;
- 3) for mind maps, certain criteria were evaluated by 5 people;
- 4) for mass open online courses, certain criteria were evaluated by 12 people.

This is a result of numerous factors. For example, during the interviews, it turned out that the vast majority of associate professors, senior lecturers, and assistants of departments who provide the educational process in the basics of programming in the preparation of future software engineers, could not give reasonable answers to determine the indicators of the criteria for using mind maps and massive open online courses in teaching the basics of programming, as they are not familiar with such technologies at all. More than 50% of them use in the educational process compilers designed for use in only one programming language.

To understand the degree of manifestation of each criterion, the respondents were asked to evaluate its indicators. The indicators were evaluated according to the following parameters: 0 points – the indicator is not observed, 1 point – the indicator is more not observed than observed, 2 points – the indicator is more observed than not observed, 3 points – the indicator is fully observed. The indicator was considered positive if the value of the corresponding coefficient – the arithmetic mean of its parameters – was at least 1.5.

Along with this, the criterion was considered insufficiently manifested if less than 50% of its indicators were positive; critical manifestation of the criterion is 50%–55%; sufficient manifestation is 56%–75%; high manifestation is 76%–100%.

4. Results

4.1. The content and structure of the selection criteria for web-oriented technologies for teaching the basics of programming to future software engineers

According to the results of the survey, the most significant tools for teaching the basics of programming were chosen:

- Web-based compilers: [codepad](#) and [ideone.com](#).
- automated systems for checking programming tasks: [Algotester](#), [NetOI](#), [Eolymp](#).
- web-oriented mind maps: [Bubbl.us](#), [Mindomo](#), [Mindmeister](#).
- massive open online courses: [Prometheus](#), [Coursera](#), [Cisco Networking Academy](#), [Udacity](#), [Udemy](#).

At the same time, the results of the survey were carried out according to formulas (1) – (4). As a result, we got $W = 0.58$ (for web-oriented learning technologies other than MOOC platforms) and $W = 0.82$ (for MOOC platforms). The obtained values are significantly different from zero, so it can be argued that there is an objective agreement between the experts and the total ranks are quite objective.

Taking into account the above features of training software engineers in higher education institutions, we will define the criteria for selecting open web-oriented technologies for teaching the basics of programming future software engineers.

In [3], the “criterion” is defined as “a standard or principle by which something is judged, or with the help of which a decision is made”. Therefore, under the *criteria for selecting web-oriented technologies for teaching the basics of programming*, we will understand such qualities, features, and properties of web-oriented technologies that are necessary for the successful teaching of the basics of programming to future software engineers.

The criteria for selecting web-oriented technologies for teaching the basics of programming to future software engineers were identified taking into account external and internal criteria and indicators of the quality of educational activities given by Ogawa and Collom [12]. It should be noted that such quality criteria are determine the degree of compliance of pedagogical activity with established goals, standards, and norms.

It is essential that the number of indicators for determining the degree of significance of the criteria for selecting web-oriented technologies for teaching the basics of programming to future software engineers may turn out to be large, and individual indicators of a certain criterion may not be significant enough for its selection. The experience of conducting experimental pedagogical research indicates the need for a reasonable limitation on the number of indicators (quite often, scientists use from 3 to 7 indicators for each criterion).

We use the following criteria and corresponding indicators for the selection of web-oriented technologies for teaching the basics of programming:

- *compilers* – design criterion (reliability; availability; free) and the functional criterion (user input data; ease of use; multilingualism);
- *automated systems for checking programming tasks* – design criterion (reliability; availability; multilingualism; ease of use; free); informational and didactic criterion (task bank; classification of tasks by sections; creation competitions; information about attempts to solve a problem; methodological section; rating; help section); communication criterion (registration of users; providing access with differentiation of access rights; communication between registered users; creation of groups).
- *mind maps* – design criterion (adaptability; free; ease of use; cloud infrastructure) and functional criterion (multilingualism; storage of mind maps; distribution of mind maps; template library);
- *massive open online courses* – informational and didactic criterion (creating your own courses; existing course bank; covering topics of different subjects; covering topics of discipline; tasks; courses of world universities); functional criterion (multilingualism; user registration; one account for several courses).

Let’s take a closer look at the results for each of the selected web-oriented technologies for teaching the basics of programming.

4.2. Web-oriented compilers

The design criterion characterizes the clarity, superiority, and accessibility of the use:

- The ‘reliability’ indicator ‘characterizes the smooth functioning of web-oriented compilers (the indicator was evaluated by a group of experts).

- The “availability” indicator assumes that the compiler should be available in the Internet, and also at any time and in any place (including both for teachers and students).
- The “free” indicator provides for a free tariff plan for use.

Table 1 shows the design criterion scores for each of the selected compilers.

Table 1

Design criterion for web-oriented compilers and its indicators.

Compilers	Indicators			Manifestation of the criterion
	Reliability	Availability	Free	
codepad	2.22	2.56	3	100%
ideone.com	2.56	2.67	3	100%

The functional criterion characterizes the functional component of the compiler:

- The “user input data” indicator provides the possibility of entering various input data when the program executed.
- The “ease of use” indicator suggests that the compiler should be easy to use for the student, it also provides for convenience and clarity in use, organization of access, and mastering the use of different groups of students.
- The “multilingualism” indicator assumes that the compiler supports different programming languages.

Table 2 shows the functional criterion scores for each of the selected compilers.

Table 2

Functional criterion for web-oriented compilers and its indicators.

Compilers	Indicators			Manifestation of the criterion
	User input data	Ease of use	Multilingualism	
codepad	0	1.56	1.56	66%
ideone.com	2.67	2.67	3	100%

So, as the study shows, ideone.com is the most convenient and high-quality toolkit among web-oriented compilers for teaching the basics of programming to future software engineers by manifesting all criteria.

4.3. Web-oriented automated systems for checking programming tasks

The design criterion characterizes the convenience, reliability, and safety of using web-oriented automated systems for checking programming tasks.

The indicators “reliability”, “availability”, “multilingualism”, “ease of use” and “free” are described above.

Table 3 presents the indicators of the design criterion for each of the selected web-oriented automated systems for checking programming tasks (WOASCPT).

Table 3

Design criterion for web-oriented automated systems for checking programming tasks and its indicators.

WOASCPT	Indicators					Manifestation of the criterion
	Reliability	Availability	Multilingualism	Ease of use	Free	
Algotester	2.17	2.42	1.5	1.67	3.00	80%
NetOI	1.92	2.17	1.75	1.83	3.00	100%
Eolymp	2.50	2.50	2.67	2.42	3.00	100%

The *informational and didactic criterion* characterizes the informational and didactic components of a web-oriented automated system for checking programming tasks:

- The “task bank” indicator characterizes the presence of a large number of tasks in web-oriented automated systems for checking programming tasks.
- The “classification of tasks by sections” indicator implies the systematization and classification of all available tasks according to different classifications [21].
- The “creating competitions” indicator is responsible for the possibility of creating competitions for participants in the educational process.
- The “information about attempts to solve the problem” indicator implies the presence of information about the number of attempts to solve a particular problem in general and the percentage for which this or that task is credited.
- The “methodological section” indicator characterizes the presence of a methodical section in a web-oriented automated system for checking programming tasks.
- The “rating” indicator assumes the presence of a general rating of users, including separately in competitions.
- The “help section” indicator suggests that a web-oriented automated system for checking programming tasks should contain a help section or at least feedback.

Table 4 presents the indicators of the information and didactic criterion for each of the selected web-oriented automated systems for checking programming tasks.

Table 4

Information and didactic criterion of web-oriented automated systems for checking programming tasks and its indicators.

WOASCPT	Indicators							Manifestation of the criterion
	Task bank	Classification of tasks by sections	Creating competitions	Information about attempts to solve the problem	Methodological section	Rating	Help section	
Algotester	1.25	0.25	2.42	2.42	0.00	2.33	2.50	57%
NetOI	1.42	1.42	2.50	0.00	0.00	0.42	1.42	14%
Eolymp	2.75	2.75	2.83	2.67	2.50	2.50	2.83	100%

The *communication criterion* characterizes the organizational and communication component of a web-oriented automated system for checking programming tasks:

- The “registration of users” indicator implies the possibility of registering new students on their own, without outside support, as well as the differentiation of access rights with different opportunities for students and teachers.
- The indicator “providing access with differentiation of access rights” provides access to the system for any user by delimiting access rights to different categories of users: students, teachers, administrators, and parents.
- The indicator “communication between registered users” implies the ability to maintain communication.
- The “creation of groups” indicator implies the possibility of creating groups for more convenient communication and notification of users, as well as the possibility of creating competitions in a specific group.

Table 5 presents the indicators of the communication criterion for each of the selected web-oriented automated systems for checking programming tasks.

Table 5

Communication criterion for web-oriented automated systems for checking programming tasks and its indicators.

WOASCP	Indicators				Manifestation of the criterion
	Registration of users	Providing access with differentiation of access rights	Communication between registered users	Creation of groups	
Algotester	1.58	1.58	0.50	0.00	50%
NetOI	0.00	1.58	0.67	0.00	25%
Eolymp	2.83	2.42	2.50	2.42	100%

So, as the study shows, Eolymp is the most convenient and high-quality tool among web-oriented systems for checking programming tasks for the manifestation of all criteria.

4.4. Web-oriented mind maps for learning the basics of programming

The design criterion characterizes convenience, reliability, and adaptability in use:

- The “adaptability” indicator characterizes the mind map in terms of adaptation to use in various operating systems (Windows, Android, iOS, etc.).
- The “free” indicator assumes a free usage plan, although not a fully functional one.
- The “ease of use” indicator suggests that the mind map should be easy to use.
- The “cloud infrastructure” indicator characterizes the cloud-oriented service for creating smart maps.

Table 6 presents the indicators of the design criterion for each of the selected mind maps.

The functional criterion characterizes the functional component of mind maps:

- The “multilingualism” indicator assumes that mind maps support different languages.

Table 6

Design criterion for web-oriented mind maps and its indicators.

Mind map	Indicators			Manifestation of the criterion
	Adaptability	Free	Ease of use	
Bubbl.us	2.4	1.4	1.6	75%
Mindomo	2.4	1.4	2.4	75%
Mindmeister	2.8	1.8	2.6	100%

- The “storage of mind maps” indicator implies the ability to store mind maps in the form of pictures.
- The “distribution of mind maps” indicator characterizes whether it is possible to distribute a mind map on the Internet.
- The “template library” indicator provides for the ability to select a template from existing ones.

Table 7 shows the indicators of the functional criterion for each of the selected mind maps.

Table 7

Functional criterion of web-oriented mind maps and its indicators.

Mind map	Indicators			Manifestation of the criterion
	Multilingualism	Storage of mind maps	Distribution of mind maps	
Bubbl.us	0.4	2.4	2.4	50%
Mindomo	2.6	2.6	2.6	75%
Mindmeister	2.8	2.8	2.6	100%

So, as the study shows, the most convenient and high-quality toolkit among web-oriented mind maps for teaching the basics of programming to future software engineers for the manifestation of all criteria is Mindmeister.

4.5. Massive open online courses for teaching the basics of programming

The *informational and didactic criterion* characterizes the informational and didactic components of the MOOC platform:

- The “creating your own courses” indicator characterizes the ability to create your own online courses as a teacher.
- The “existing course bank” indicator provides a large number of courses.
- The “covering topics of different subjects” indicator ensures the availability of courses in various academic disciplines, provided by the curriculum for the preparation for bachelors in software engineering.
- The “covering topics of discipline” indicator provides for the coverage of topics of the academic discipline according to the curriculum and work program of the discipline.

- The “tasks” indicator characterizes the tasks that are available after each video, or at least questions to test the consolidated knowledge.
- The “courses of world universities” indicator provides for the availability of courses in various academic disciplines offered by world universities.

Table 8 presents the indicators of the information and didactic criterion for each of the selected MOOC platforms.

Table 8

Information and didactic criterion of MOOC platforms and its indicators.

MOOC	Indicators						Manifestation of the criterion
	Creating your own courses	Existing course bank	Covering topics of different subjects	Covering topics of discipline	Tasks	Courses of world universities	
Prometheus	0.00	2.50	2.50	2.58	2.50	1.67	83%
Coursera	0.00	2.50	2.33	2.50	2.58	3.00	83%
Cisco Networking Academy	0.00	2.17	1.50	2.50	2.50	3.00	83%
Udacity	0.00	2.75	2.58	2.42	1.50	0.00	66%
Udemy	2.67	3.00	2.58	2.50	2.58	1.50	100%

The functional criterion covers the functionality of massive open online courses:

- The “multilingualism” indicator characterizes whether the MOOC has the opportunity to listen to the course in different languages.
- The indicator “user registration” characterizes whether the MOOC is open to all or only to registered users.
- The indicator “one account for several courses” characterizes the ability to register under one name for several courses.

Table 9 presents the functional criterion indicators for each of the selected MOOC platforms.

Table 9

Functional criterion of MOOC platforms and its indicators.

MOOC	Indicators			Manifestation of the criterion
	Multilingualism	User registration	One account for several courses	
Prometheus	1.33	2.83	3.00	67%
Coursera	1.17	2.67	2.33	67%
Cisco Networking Academy	1.92	1.33	1.25	33%
Udacity	1.42	2.50	2.67	67%
Udemy	2.50	2.50	2.67	100%

As the study shows, the most convenient and high-quality tool among the MOOC platforms for use in the training of software engineers in terms of the level of manifestation of all criteria is the Udemy MOOC platform.

5. Discussion

As you can see, experts identify a number of web-oriented learning technologies to improve the process of teaching the basics of programming to future software engineers. In the process of training software engineers, all specified learning tools that meet the selected criteria and indicators should be used. However, given the constant changes in the field of ICT, teachers should review the proposed technologies once a year and, in the case of fundamental changes in functionality or the introduction of new specified teaching technologies, conduct an updated survey to establish the best quality tools that are appropriate for use in the software engineering training.

The use of these technologies for teaching the basics of programming not only facilitates the educational process but also provides students with various technologies that will be useful to them in their future professional activities.

It should be noted that the use of MOOCs in software engineer training is only used if a portion of the content is pulled out for individual study, which, according to an analysis of curriculums for training software engineers, is at least 66%. They are simple to use and can be used as an additional learning tool.

6. Conclusion and future work

Inadequate awareness of the problem of applying cloud-oriented and web-oriented learning tools, as well as a lack of examples of their practical application, contributed to a complete examination of the problem of their use in the HEI educational process. To that purpose, it proved necessary to examine the available cloud-based and web-based learning tools for use in the training of software engineers. As a result, it was determined that for the selection of cloud-oriented and web-oriented learning tools, it is advisable to use software engineers for training, certain criteria and indicators should be taken into account.

It was established which of the proposed web-oriented technologies best fit certain criteria for teaching the basics of programming to future software engineers. In particular, among web-oriented compilers for teaching the basics of programming to future software engineers, ideone.com should be recommended for use, among web-oriented systems for checking programming tasks – the Eolymp Internet portal, and among web-oriented mind maps – Mindmeister, among the MOOC platforms for use in training software engineers for the manifestation of all criteria is the MOOC platform Udemy.

In the future, the essential components of a methodological system for teaching the basics of programming to future software engineers will need to be developed.

References

- [1] Abdullahi, M.S.I., Salleh, N., Nordin, A. and Alwan, A.A., 2018. Cloud-based learning system for improving students' programming skills and self-efficacy. *Journal of Information and Communication Technology*, 17(4), pp.629–651. Available from: <https://e-journal.uum.edu.my/index.php/jict/article/view/jict2018.17.4.2642>.
- [2] Bykov, V.Y., Spirin, O.M., Biloshchytskyi, A.O., Kuchansky, A.Y., Dikhtiarenko, O.V. and Novytskyi, O.V., 2020. Open digital systems for assessment of pedagogical research results. *Information technologies and learning tools*, 75(1), p.294–315. Available from: <https://doi.org/10.33407/itlt.v75i1.3589>.
- [3] criterion noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com, 2022. Available from: <https://www.oxfordlearnersdictionaries.com/definition/english/criterion>.
- [4] Galchevska, G., 2015. Using the international scientometric databases of open access in scientific research. *Journal of Information Technologies in Education (ITE)*, (23), pp.115–126. Available from: <https://doi.org/10.14308/ite000541>.
- [5] Glazunova, O.G., Parhomenko, O.V., Korolchuk, V.I. and Voloshyna, T.V., 2021. The effectiveness of GitHub cloud services for implementing a programming training project: students' point of view. *Journal of Physics: Conference Series*, 1840(1), p.012030. Available from: <https://doi.org/10.1088/1742-6596/1840/1/012030>.
- [6] Gordiyenko, T., Velychko, O. and Salceanu, A., 2018. The Expert's Competence Evaluation in Electrical Engineering Education. *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*. pp.0011–0016. Available from: <https://doi.org/10.1109/ICEPE.2018.8559852>.
- [7] Holovnya, O., 2015. Criteria for selecting virtualization software in teaching UNIX-like operating systems. *Journal of Information Technologies in Education (ITE)*, (24), pp.119–133. Available from: <https://doi.org/10.14308/ite000548>.
- [8] Kremen, V., ed., 2021. *National Report on the State and Prospects of Education Development in Ukraine*. Kyiv: KONVI PRINT. Available from: <https://doi.org/10.37472/NAES-2021-en>.
- [9] Küçük, K., Bayılmış, C. and Msongaleli, D.L., 2021. Designing real-time IoT system course: Prototyping with cloud platforms, laboratory experiments and term project. *The International Journal of Electrical Engineering & Education*, 58(3), pp.743–772. Available from: <https://doi.org/10.1177/0020720919862496>.
- [10] Liao, J., Chen, S. and Xiong, H., 2017. A cloud-based online coding platform for learning coding-related courses of computer science. *ICIC Express Letters, Part B: Applications*, 8(1), p.109 – 116.
- [11] Markova, O.M., Semerikov, S.O., Striuk, A.M., Shalatska, H.M., Nechypurenko, P.P. and Tron, V.V., 2019. Implementation of cloud service models in training of future information technology specialists. *CTE Workshop Proceedings*, 6, pp.499–515. Available from: <https://doi.org/10.55056/cte.409>.
- [12] Ogawa, R. and Collom, E., 1998. *Educational Indicators: What Are They? How Can Schools and School Districts Use Them?* (SO-004). Riverside: California Educational Research Cooperative. Available from: <https://eric.ed.gov/?id=ED432811>.
- [13] Panchenko, L. and Khomiak, A., 2020. Education statistics: Looking for a case-study

- for modelling. *CEUR Workshop Proceedings*, 2732, pp.948–963. Available from: <http://ceur-ws.org/Vol-2732/20200948.pdf>.
- [14] Rassoavytska, M.V. and Striuk, A.M., 2018. The system of cloud-oriented tools of learning computer science disciplines of engineering specialties students. *CTE Workshop Proceedings*, 5, pp.20–26. Available from: <https://doi.org/10.55056/cte.131>.
- [15] Segrelles Quilis, J.D., Moltó, G. and Blanquer, I., 2021. A cloud framework for problem-based learning on grid computing. *Journal of Parallel and Distributed Computing*, 155, pp.24–37. Available from: <https://doi.org/10.1016/j.jpdc.2021.04.012>.
- [16] Spirin, O.M., Ivanova, S.M., Iatsyshyn, A.V., Luparenko, L.A., Dudko, A.F. and Kilchenko, A.V., 2020. The model for the application of open electronic scientific and educational systems to the development of researchers' information and research competence. *Information technologies and learning tools*, 77(3), p.302–323. Available from: <https://doi.org/10.33407/itlt.v77i3.3985>.
- [17] Tkachuk, V., Semerikov, S., Yechkalo, Y.V., Khotskina, S. and Soloviev, V.N., 2020. Selection of Mobile ICT for Learning Informatics of Future Professionals in Engineering Pedagogy. In: O. Sokolov, G. Zholtkevych, V. Yakovyna, Y. Tarasich, V. Kharchenko, V. Kobets, O. Burov, S. Semerikov and H. Kravtsov, eds. *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kharkiv, Ukraine, October 06-10, 2020*. CEUR-WS.org, *CEUR Workshop Proceedings*, vol. 2732, pp.1058–1068. Available from: <http://ceur-ws.org/Vol-2732/20201058.pdf>.
- [18] Tryus, Y. and Kachala, T., 2015. Cloud technologies in management and educational process of Ukrainian technical universities. *Journal of Information Technologies in Education (ITE)*, (19), pp.022–033. Available from: <https://doi.org/10.14308/ite000481>.
- [19] Tseng, C.H., Liu, J.S., Chen, Y.H., Hui, L., Jiang, Y.R. and Lin, J.R., 2018. The Requirement Analysis and Initial Design of a Cloud and Crowd Supported Mathematics Learning Environment for Computer Science Students. In: N.Y. Yen and J.C. Hung, eds. *Frontier Computing*. Singapore: Springer Singapore, pp.219–226. Available from: https://doi.org/10.1007/978-981-10-3187-8_23.
- [20] Vakaliuk, T.A., 2017. Structural and functional model of cloud oriented learning environment for bachelors of informatics training. *Information technologies and learning tools*, 59(3), p.51–61. Available from: <https://doi.org/10.33407/itlt.v59i3.1674>.
- [21] Vakalyuk, T.A. Solution creative programming tasks of future teachers informatics. *Visnyk Chernihivskoho natsionalnoho pedahohichnoho universytetu imeni T. H. Shevchenka*, (113), pp.109–114. Available from: <http://eprints.zu.edu.ua/10305/>.
- [22] Zhao, J. and Forouraghi, B., 2013. An Interactive and Personalized Cloud-Based Virtual Learning System to Teach Computer Science. In: J.F. Wang and R. Lau, eds. *Advances in Web-Based Learning – ICWL 2013*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.101–110. Available from: https://doi.org/10.1007/978-3-642-41175-5_11.