CZU: 37.016:004:373.5 **https://doi.org/10.59295/sum9(179)2024_06**

MONITORING THE COMPUTER SCIENCE CURRICULUM IN HIGH SCHOOLS: ADAPTATION AND EFFICIENCY IN THE CONTEMPORARY EDUCATIONAL CONTEXT

Maria MARIN, Ghenadie MARIN,
State University of Moldova

Lilia IVANOV,
Ion Creangă Theoretical High School, Coșnița village, Dubăsari district

Monitoring the computer science curriculum involves a continuous process of assessment, adaptation, and improvement, ensuring that the content delivered to students is not only up-to-date but also meets the demands of the modern technological world. This process includes evaluating the implementation of new technologies, teaching methodologies, and assessment tools, while also considering how these innovations impact student learning outcomes.

Through effective monitoring, it is possible to identify areas of improvement and gaps in the curriculum that need addressing, while also recognizing the strengths of existing educational practices. Feedback from educators, students, and stakeholders is crucial in shaping a curriculum that is both efficient and responsive to the demands of the digital age.

The monitoring of the computer science curriculum at the secondary school level is essential for adapting to the needs of modern education. This continuous process ensures that students are equipped with the necessary skills to succeed in a digitalized world and that educators are supported in creating dynamic, relevant, and effective learning environments.

Keywords: *education, monitoring, teaching, technology, informatics.*

MONITORIZAREA CURRICULUMULUI DE INFORMATICĂ ÎN LICEE: ADAPTARE ȘI EFICIENȚĂ ÎN CONTEXTUL EDUCAȚIONAL CONTEMPORAN

Monitorizarea curriculum-ului de informatică implică un proces continuu de evaluare, adaptare și îmbunătățire, asigurându-se că conținutul livrat studenților nu este doar actualizat, ci și satisface cerințele lumii tehnologice moderne. Acest proces include evaluarea implementării noilor tehnologii, metodologii de predare și instrumente de evaluare, luând în considerare și modul în care aceste inovații afectează rezultatele învățării elevilor.

Printr-o monitorizare eficientă, este posibil să se identifice domeniile de îmbunătățire și lacunele din curriculum care trebuie abordate, recunoscând în același timp punctele forte ale practicilor educaționale existente. Feedback-ul din partea educatorilor, studenților și părților interesate este esențial în formarea unui curriculum care este atât eficient, cât și receptiv la cerințele erei digitale.

Monitorizarea curriculum-ului de informatică la nivelul gimnaziului este esenţială pentru adaptarea la nevoile învăţământului modern. Acest proces continuu asigură că studenții sunt echipați cu abilitățile necesare pentru a reuși într-o lume digitalizată și că educatorii sunt sprijiniți în crearea unor medii de învățare dinamice, relevante și eficiente.

Cuvinte-cheie: *educație, monitorizare, predare, tehnologie, informatică.*

Introduction

Education is in a state of continuous transformation, a process amplified by rapid technological advancement and the social and economic changes that shape societal needs. In this context, the analysis and monitoring of the computer science curriculum become essential to ensure the relevance and effectiveness of the educational process. The management of educational innovations in the field of computer science involves identifying, implementing, and managing necessary changes to address new challenges in education and to enhance students' preparation for the demands of a digital society.

Monitoring the computer science curriculum is essential for evaluating the impact of new methods and technologies in computer science education, playing a significant role in improving the quality of education, institutional performance, and students' preparedness to adapt to the new realities of the labor market. Through effective management of innovations, educational institutions can ensure high-quality education and respond swiftly to technological and pedagogical changes.

In this context, the management of educational innovations in computer science does not only mean the implementation of new technologies or changes in teaching methods, but also the continuous evaluation of their impact on the educational process and the assurance of the sustainability of these changes. Evaluating the computer science curriculum allows for its continuous adjustment to ensure alignment with technological developments and students' needs.

For the educational system of the Republic of Moldova, the analysis and monitoring of the computer science curriculum are increasingly important in light of the challenges faced by schools, such as adapting to emerging technologies and integrating them into the educational process. Implementing effective strategies for managing innovations is crucial for promoting the continuous development of education, with the computer science curriculum playing a central role in this transformation.

As Drucker (1985) emphasized, innovations in education involve the adoption of new approaches and technologies that meet the current needs of students. In this regard, computer science teachers are the key actors in implementing these innovations, bearing the responsibility of integrating modern technologies into their lessons and adapting these tools to students' specific needs [1]. Therefore, curriculum monitoring is a crucial tool to evaluate how effectively these technologies are being used and to identify any gaps or areas that require improvement.

Furthermore, managing educational innovations within computer science involves creating a favorable environment for the development and implementation of innovative ideas. This environment can be supported through an organizational culture that encourages change and through the active involvement of all stakeholders in the school community – teachers, students, and parents – in the process of curriculum development and evaluation. Continuous monitoring of the implementation of these innovations allows for adjustments based on feedback from students and teachers, thereby contributing to the improvement of the learning process.

Innovations in computer science education are not limited to the technologies used but also extend to teaching methodologies, student assessment, and the organization of educational activities. For instance, new learning methodologies, such as project-based or collaborative learning, can be effectively integrated into the computer science curriculum to stimulate critical thinking and problem-solving skills. In this regard, monitoring and analyzing the curriculum help identify best teaching practices and assess their effectiveness.

Another important aspect of the analysis of the computer science curriculum is the creation of a more inclusive and equitable educational environment. By managing educational innovations, educational institutions can ensure that all students, regardless of their social background or abilities, have access to quality education. Curriculum monitoring thus helps identify and correct inequities or barriers in access to educational resources, ensuring that each student benefits from personalized learning.

Adopting an effective management of innovations in computer science education can contribute to the creation of excellent educational institutions that better meet students' needs and enhance their performance. Constant evaluation of the curriculum allows for the adjustment of educational strategies and the development of innovative solutions that improve the learning experience. Collaboration between teachers, students, and parents is essential for the success of this management, and curriculum monitoring becomes a key tool in the implementation and evaluation of changes.

Aspects of monitoring the computer science curriculum

In a world characterized by rapid technological advancement and swift changes in the labor market, monitoring the computer science curriculum becomes an essential priority to ensure relevant and high-quality education. This process involves not only the periodic updating of content and teaching methods but also the continuous evaluation of their effectiveness. Thus, educational institutions can adapt their programs to the demands and challenges of a digitized society, preparing students for future professional requirements.

1. ***The importance of monitoring the computer science curriculum.*** The computer science curriculum must address complex and dynamic needs, including both fundamental technical skills and socio-emotional abilities, such as critical thinking and creativity. Monitoring this curriculum does not only involve implementing new technologies and teaching methods but also continuously assessing their impact on students' outcomes. An effective computer science curriculum must prepare students not only for exams but also to become active citizens, capable of adapting to the continuous changes in technology and society.

2. ***Integration of Emerging Technologies.*** A key aspect of monitoring the computer science curriculum is the integration of emerging technologies, such as e-learning platforms, the use of mobile devices, and online collaboration tools. These technologies provide new learning opportunities, flexibility, and accessibility, transforming how students interact with information and teachers. By monitoring the effectiveness of these technologies in the educational process, institutions can adjust the curriculum to maximize their positive impact on learning and develop students' digital skills.

3. ***Adapting content to societal changes.*** Given the rapid pace of technological and cultural changes, the computer science curriculum must be flexible and adaptable. Vladimir Guțu, an educational expert, emphasizes that schools must evolve alongside society to maintain the relevance and currency of the teaching content. This requires periodic review of the curriculum and the adoption of modern teaching-learning methods that stimulate students› interest and prepare them for success in an ever-changing world.

4. ***Involvement of the educational community.*** Monitoring the computer science curriculum involves active collaboration between teachers, students, parents, and other stakeholders in the educational community. Research has shown that involving these stakeholders can enhance the efficiency of implementing educational innovations and help identify specific challenges and needs within each community. Continuous feedback from this community is a valuable tool for adjusting and optimizing the curriculum.

5. ***Teaching efficiency and quality of education.*** A central component of monitoring the curriculum is promoting teaching efficiency and ensuring the quality of education. By adopting innovative and interactive methodologies, teachers can improve the knowledge assimilation process and stimulate the development of students' critical and creative skills. Constant monitoring of how these methodologies affect students' performance is essential to ensure quality education tailored to contemporary needs.

6. ***Evaluating the impact of educational innovations***. Continuous evaluation of the impact of educational innovations on learning is a fundamental aspect of monitoring the curriculum. By analyzing students' performance, feedback from teachers, and case studies, valuable data can be obtained on the effectiveness of new teaching strategies and integrated technologies. These evaluations allow for adjustments to be made to the methodologies, ensuring they remain relevant and effective in developing students' competencies.

7. ***Preparing students for the future.*** A well-monitored computer science curriculum aims to prepare students for the current and future demands of the labor market and society. In an era of rapid change, digital skills and adaptability become essential abilities. By adapting the curriculum to new realities and monitoring its impact, educational institutions can contribute to the development of active citizens who are prepared to face the challenges of a digitized world.

Characterization of the specifics of the computer science curriculum in secondary education

The curriculum for the computer science subject in secondary education (grades VII-IX) in the Republic of Moldova is regulated by the Framework plan for primary, secondary, and high school education (academic year 2019-2020) and the disciplinary curriculum for computer science, approved by the Ministry of Education, Culture, and Research. This foundational document specifies the essential competencies that students need to develop, the mandatory content to be studied, and methodological suggestions for effective teaching. Structured by school year, the curriculum provides a unified framework for planning teaching activities for this subject.

The Framework Plan sets the number of mandatory hours for computer science in secondary education as follows:

– Grade 7: 1 hour per week (35 hours/year);
– Grade 8: 1 hour per week (35 hours/year);
– Grade 9: 1 hour per week (35 hours/year).

For secondary education, the curriculum focuses on four main components:

1. Representation of Information: This component aims to develop competencies related to the encoding and decoding of information in various encoding systems (e.g., binary, hexadecimal, ASCII), the representation of information in different forms (text, images, sound, video), the use of positional numeral systems, and performing conversions between these systems.

Pupils are expected to understand fundamental concepts such as bits and bytes, encode and decode simple messages, and recognize file formats, associating them with the type of information represented.

In Grade 7, for instance, students learn about binary and hexadecimal numeral systems, perform conversions with the decimal system, and use the ASCII code to encode simple messages.

2. Information Processing: Students are encouraged to develop algorithmic thinking and basic programming skills by learning to analyze problems, decompose complex tasks into simple steps, represent algorithmic solutions using diagrams or pseudocode, and implement them in visual programming environments [2].

In Grade 7, students familiarize themselves with introductory programming concepts, such as control structures (conditional and repetitive instructions), variables, and basic data types, representing algorithms through simple programs and pseudocode. They also acquire practical programming skills in visual environments such as Scratch.

In Grade 8, programming skills are deepened, with students being introduced to concepts like subprograms (functions and procedures), structured data types (strings, lists/arrays), and reading/writing data to text files. Recommended programming environments include Python and Pascal.

3. Communication of Information: This component focuses on developing competencies related to the use of electronic communication services and applications (such as email and instant messaging), creating and publishing web content, as well as adhering to online security and ethical standards [2].

Students are trained to use electronic communication tools effectively and responsibly, create and publish simple web content (static web pages), and understand the risks associated with the online environment, adopting ethical and secure behavior.

In Grade 7, students study introductory concepts about the internet and web services, learn how to use email and instant messaging, and how to search for and select relevant information online.

In Grade 8, their competencies are extended to include creating and publishing simple web pages using HTML, adhering to security standards (such as strong passwords and personal data protection), and online ethics (including respect for copyright).

4. Interdisciplinary integration of information technologies: This component focuses on developing the competencies necessary for the integrated use of ICT tools in learning activities across various fields, promoting responsible and efficient use of technologies for research, communication, collaboration, and learning [2].

Students are encouraged to understand the importance of information technologies in different fields and use them appropriately in their learning activities. They will acquire skills in using ICT for research, information processing, communication, and collaboration in interdisciplinary contexts.

In Grade 7, students learn to use ICT for simple interdisciplinary projects, collaborate online for task completion, and respect copyright when using digital resources.

In Grade 8, the competencies for ICT integration are deepened through more complex interdisciplinary projects, involving a variety of ICT tools for research, information processing, communication, and collaboration.

Specific competencies in the informatics curriculum for middle school

The curriculum focuses on developing basic digital competencies essential for the active integration of students into modern information society. Thus, according to the Informatics Curriculum, the goals of studying informatics at the middle school level are presented as follows:

1. *Development of digital competencies necessary for active integration of students into the information society*:

- Responsible and effective use of information and communication technologies (ICT) in various life and activity contexts;
- Development of logical and algorithmic thinking through describing, analyzing, and solving problems using computers;
- Solving practical problems from various fields using computers and software applications;
- Creation of simple digital products (documents, presentations, web pages, etc.) for communication and dissemination of information;
- Effective communication and collaboration in virtual environments, respecting security and ethical standards.

2. *Development of analysis, synthesis, generalization, and transfer skills of knowledge and competencies acquired in informatics in various life and activity contexts, including learning other school subjects*.

3. *Development of a positive, responsible, and critical attitude toward information and communication technologies, awareness of their impact on personal, social, and professional life.*

4. *Cultivation of interest and motivation for the study and application of knowledge in the field of informatics for lifelong learning, in accordance with the requirements of the information society.*

5. *Development of key transdisciplinary competencies:*

- Learning competencies (organizing one's own learning, problem-solving, critical thinking, etc.);
- Communication competencies in romanian and foreign languages;
- Competencies in using ICT;
- Collaboration and teamwork competencies.

6. *Preparation of students for continuing studies in the field of information and communication technologies at the next educational levels (high school, higher education), as well as for integration into the labor market as competent users of information and communication technologies (ICT)*.

These objectives include both the development of specific digital competencies and the formation of transversal skills, as well as logical and critical thinking, which are essential for the active integration of students into the informational society. Additionally, the curriculum aims to stimulate students' interest and motivation for continuing their studies in the field of ICT.

A detailed analysis of the specific competencies outlined in the IT curriculum for grades VII-IX provides a clear understanding of their progression throughout the middle school cycle, in accordance with the taxonomy presented in the curriculum. The curriculum is structured so that competencies develop gradually as students progress through middle school, ensuring a constant progression from basic levels to advanced competencies.

*Competencies in grade VII: understanding and application***.** In grade VII, the focus is on medium-level competencies in the taxonomy, centered on understanding and application. Students begin to explore coding systems and develop their skills in encoding and decoding information in specific formats such as binary, hexadecimal, or ASCII. These activities aim to provide them with a deeper understanding of how information is processed digitally.

Additionally, students begin to describe and represent basic algorithms using flowcharts, simple programs, or pseudocode, which allows them to structure their logical thinking and understand the essential principles of algorithms. Thus, at this level, students acquire essential skills for developing a solid foundation in computer science.

Competencies in grade VIII: analysis and creation. In grade VIII, the curriculum expands its objectives to include more advanced competency levels, such as analysis and creation. Students begin implementing algorithms in programming languages such as Pascal and applying previous knowledge to more complex problems. They are also encouraged to analyze and compare different programming solutions, evaluating them in terms of efficiency, complexity, and scalability.

Through these activities, students gain essential analytical competencies and begin to develop critical thinking, which is necessary for solving computer science problems and evaluating technical solutions.

Thus, at this stage, the curriculum encourages students to be innovative and to develop their problem-solving skills efficiently.

Competencies in grade IX: evaluation and creation. At the grade IX level, the competencies outlined in the curriculum reach the highest levels of the taxonomy, focusing on evaluation and actual creation. Students begin working with more complex data structures, such as non-linear structures, and applying advanced processing algorithms. These activities require a high level of synthesis and analysis, preparing students for a deep understanding of computer science.

Students also develop software applications using programming languages and are encouraged to evaluate and optimize the performance of these applications, which allows them to acquire advanced critical evaluation skills. This component of the curriculum helps students refine both their programming skills and the critical thinking necessary for developing efficient and sustainable solutions in the current technological context.

This coherent and progressive structure, aligned with the taxonomy of specific computer science competencies, ensures thorough preparation for students. By systematically addressing competencies from understanding and application to analysis and creation, the curriculum contributes to the development of essential digital skills. Students learn to analyze and create innovative solutions -competencies that are increasingly necessary in a technology-oriented society.

Analysis of textbooks and teaching aids for computer science at the middle school level

Textbooks and teaching aids are essential resources for the development of computer science competencies at the middle school level, providing concrete support for implementing curricular objectives and developing specific skills. These resources facilitate the progressive understanding of computer science concepts and practices, directly impacting the development of algorithmic thinking and digital competencies. In the following, we will analyze the computer science textbooks for grades VII-IX, published by Editura Știința in 2020, examining the structure, content, and activities included in each textbook.

The *computer science textbook for grade VII* guides students in taking their first steps in studying computer science, providing them with a solid foundation in algorithms and programming languages. Starting with fundamental concepts such as algorithms and their representation methods, the textbook helps students develop algorithmic thinking through practical examples inspired by everyday life, such as sorting numbers or organizing a shopping list. These activities bridge the gap between theory and real-world applications, making the information more accessible and relevant to students.

The textbook places a special emphasis on problem-solving through simple algorithms, initially using pseudocode, a language accessible to beginners. A common example is creating an algorithm to calculate the arithmetic mean of three numbers, an exercise that develops both logical thinking and the ability to structure data. Such activities contribute to the development of analysis and problem-solving skills, which are essential in the field of computer science.

As they progress, students begin to familiarize themselves with the Pascal programming language, known for its ease of use for beginners. The textbook includes clear examples that cover basic structures such as variable declarations and mathematical operations. As students advance, the exercises become more complex and involve implementing algorithms in Pascal. From a simple program that displays "Hello, World!" to a program that checks the parity of a number, students gradually develop their programming skills and enhance their logical thinking

The *eighth-grade computer science textbook* continues the development of previously acquired skills, providing a deeper understanding of data structures and algorithms. Students expand their knowledge by working with lists and matrices, learning to apply sorting and searching algorithms, which are essential for data management. For example, the textbook includes an exercise on the bubble sort algorithm, which helps students understand how to order data and how to evaluate the efficiency of an algorithmic solution.

The eighth-grade computer science textbook continues the development of previously acquired skills, providing a deeper understanding of data structures and algorithms. Students expand their knowledge by working with lists and matrices, learning to apply sorting and searching algorithms, which are essential for data management. For example, the textbook includes an exercise on the bubble sort algorithm, which helps students understand how to order data and how to evaluate the efficiency of an algorithmic solution.

Another important aspect is the in-depth exploration of advanced control structures, such as „for” and „while” loops. The textbook offers practical exercises, such as a program to calculate the factorial of a number using a “while” loop. This exercise reinforces knowledge about repetitive structures and allows students to explore the application of algorithms in more complex ways, contributing to a deeper understanding of how to automate repetitive tasks in programming.

The textbook also introduces the concept of functions, which are essential in any modern programming language. Students are encouraged to create their own functions to tackle more complex problems, thus developing their ability to decompose a problem into subproblems. For example, one exercise may ask them to write a function that returns the sum of even numbers in a list, stimulating their analytical thinking and ability to organize code. These activities offer valuable hands-on experience, preparing students for the advanced challenges of programming.

Therefore, the eighth-grade computer science textbook provides a crucial transition from basic concepts to more advanced approaches, consolidating and expanding students’ knowledge in algorithms, data structures, and programming. Through varied and challenging activities, the textbook not only develops logical thinking and analytical skills but also prepares students to confidently tackle complex problems.

The *ninth-grade computer science textbook* deepens students’ knowledge, focusing on programming and data management. Structured into four units, the textbook reinforces previous concepts and introduces new advanced topics. One of the units, dedicated to data structures and fundamental algorithms, includes concepts such as stacks, queues, and linked lists, explained through practical analogies, such as a stack of plates to illustrate how a stack works. This approach helps make abstract concepts more accessible, preparing students to implement them in code

The programming content in Pascal offers progressive exercises leading to the development of a comprehensive project. One example is creating a “Car” class, with attributes like “make,” “model,” and “year of manufacture,” teaching students to create methods that simulate functions of the car, such as starting and stopping the engine. These activities develop their technical skills while also stimulating creativity by applying principles in varied contexts. Additionally, the textbook introduces the concept of code reuse, which is essential in modern programming.

The unit on databases and SQL is presented through a practical case study, such as managing an online store. Students learn to create tables, insert data, and formulate simple queries, with an example being a database for managing a library with tables for books and authors. These activities help students understand not only the storage and retrieval of data but also how to use it efficiently.

In conclusion, the computer science textbooks for grades VII, VIII, and IX are designed to support students’ progress in developing computer skills through a gradual approach, from introductory concepts to advanced topics such as data structures and object-oriented programming. However, it is concerning that these textbooks do not fully cover all the units included in the computer science curriculum, leaving room for further exploration.

In the Republic of Moldova, supplementary teaching materials complement these textbooks, providing additional resources to support both teachers and students in deepening and expanding their knowledge.

Teaching aids are essential tools that help consolidate and deepen the knowledge acquired during computer science lessons. In the Republic of Moldova, alongside official textbooks, teachers and students have access to a variety of supplementary materials that enhance the learning experience by offering additional examples, practical exercises, and group projects. These materials are used to complement the formal learning process, either in the form of problem collections or through interactive digital resources.

A notable example is the problem collections, which are useful for consolidating algorithmic skills. For instance, a collection might include problems ranging from simple sorting and searching algorithms to more advanced exercises, such as data management in a real-world context. These activities allow students to experiment with various problem-solving strategies and test the efficiency of solutions in a controlled environment.

Worksheets are another useful type of supplementary material, often available in digital format, containing exercises that stimulate students' creativity and critical thinking. For example, an exercise might ask students to develop an algorithm for organizing a school agenda using lists and dictionaries. This type of activity encourages students to apply their knowledge in diverse contexts, fostering flexible and innovative thinking.

Additionally, many schools in the Republic of Moldova use educational digital platforms that complement textbooks, providing students with access to interactive resources such as programming simulators and online tests. One example is the E-School platform, which allows students to test their knowledge and follow additional learning modules. Through these digital resources, students can practice their programming skills interactively, receiving immediate feedback.

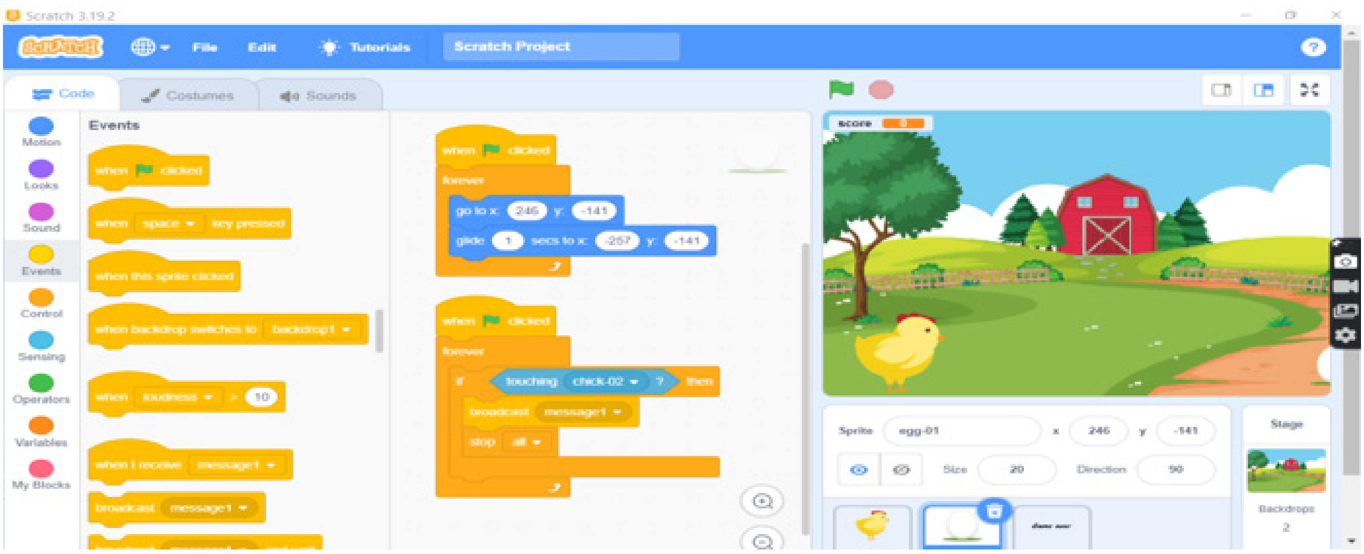
Teaching aids are also extremely helpful for preparing for exams and computer science olympiads. Specialized guides provide complex problems and detailed solutions, helping students develop their algorithmic thinking and familiarize themselves with the types of problems encountered in competitions. For example, students may be challenged to develop an algorithm for optimizing transport routes, a theme commonly found in programming contests.

For teachers, methodological guides provide support through examples of lesson plans, suggestions for integrating technology into the classroom, and strategies for personalizing instruction. For example, such a guide may recommend collaborative projects where students work together to develop a simple application, an effective method for stimulating collaborative learning and developing social skills.

With the advancement of digital technologies, various educational software and interactive applications have been developed that allow the simulation of computer science processes and concepts. These tools become valuable teaching aids in the learning process, as they enable students to apply theoretical concepts in a practical manner and develop technical skills. However, to maximize their effectiveness, it is essential that the software be supported by clear user guides and well-structured educational activities.

A significant example is „Scratch," a visual programming environment that allows students to create games, animations, and interactive stories. It is an ideal platform for developing algorithmic thinking and programming skills, being easy to use even for beginners. One of the defining features of „Scratch" is its visual interface, where colored code blocks represent different commands and programming structures. Through simple „drag and drop" of code blocks into the workspace, students can create programs without the need to write text-based code. This approach is beneficial for beginners because it eliminates the complexity of syntax and allows greater focus on the logic and structure of the program.

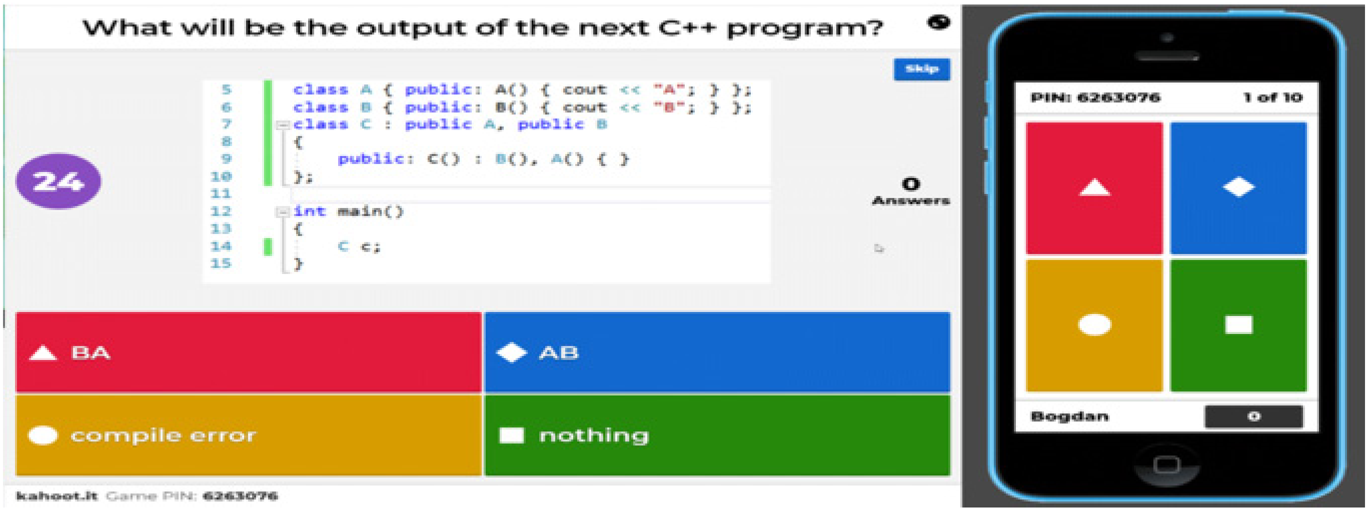
Figure 1. Scratch programming environment.



Source: [3].

Another very popular software is „Kahoot”, an educational platform widely used in schools in the Republic of Moldova, which makes learning more interactive and engaging. It allows teachers to create tests, quizzes, and games, which students can solve in real time using mobile devices or computers in the classroom. Based on the principle of gamification, the platform incorporates game elements such as points, leaderboards, and rewards into educational activities. These features stimulate friendly competition and encourage students to actively participate in lessons, as illustrated in figure 2.

Figure 2. Educational platform Kahoot.



Source: [4].

One of the main benefits of the „Kahoot” platform is its ease of use for both teachers and students. Teachers can quickly create customized quizzes on topics such as algorithms, data structures, or programming languages using an intuitive editor. Additionally, they have access to a wide range of quizzes already created by teachers worldwide, providing valuable and immediately available educational resources.

In computer science lessons, „Kahoot” is often used to assess students’ knowledge in an interactive and motivating way. For example, after learning a topic about algorithms, a teacher can create a „Kahoot” quiz with multiple-choice questions, images, or videos related to sorting or searching algorithms. Students answer the questions using their mobile devices, and the results are displayed in real-time on a screen, along with a leaderboard. This form of assessment not only measures students’ knowledge but also motivates them to improve their performance through competition with their peers.

Therefore, educational resources for computer science in secondary schools in the Republic of Moldova are diverse, including exercise books, methodological guides, educational software, online platforms, and problem collections. However, it is necessary to promote and disseminate these resources more widely, and the development of interactive digital materials and virtual simulations specific to this field would significantly contribute to improving the educational process.

Challenges in implementing the computer science curriculum at the secondary school level

The current computer science curriculum for secondary schools in the Republic of Moldova represents a significant initiative to align education with the requirements of a digitalized society. Structured around specific competencies, it facilitates a practical and applied approach to essential subjects, including current topics such as programming, robotics, and cyber security. Additionally, the curriculum emphasizes practical activities and projects aimed at developing students’ computational thinking and digital competencies.

However, the effective implementation of the curriculum faces a number of challenges.

First, the insufficient technological infrastructure in many schools limits the ability to conduct practical activities. The lack of appropriate equipment—such as computers, educational robots, and specialized software—hinders the interactive conduct of lessons.

Second, the limited number of hours allocated to the subject, especially in grades VII and VIII, constrains the comprehensive coverage of the competencies included in the curriculum, which can lead to superficial treatment of essential topics.

Another challenge is the unequal preparation of teachers for teaching computer science. While there are well-trained educators, others may face difficulties in using modern technologies or interactive teaching methods. Therefore, continuous professional development is necessary to update their digital and methodological competencies, alongside the periodic revision of curricular content to keep pace with rapid technological advancements and the needs of a knowledge-based economy.

School textbooks also need improvements. Some do not include enough practical activities and exercises to support the development of the skills outlined in the curriculum, which is why it is recommended to develop supplementary teaching materials, such as educational software and interactive online resources.

To fully harness the potential of the informatics curriculum, an integrated approach is needed: *revising the allocation of hours, properly equipping schools, continuous teacher training, and disseminating interactive digital resources*. Only through sustained collaboration between authorities, teachers, parents, and the community can the development of the digital skills necessary for students in today's society be ensured. Investment in digital education is a strategic priority, preparing students to meet future challenges and actively contribute to the progress of a knowledge-based and innovative society.

General conclusions

In conclusion, the analysis and monitoring of the computer science curriculum are essential for maintaining an effective and up-to-date education in the face of emerging technological challenges. Educational innovation management involves continuous adaptation to the needs of students and society by integrating best practices and modern technologies into the educational process. This is a dynamic approach to continuous improvement, contributing to the formation of students as responsible citizens, prepared for the future.

Monitoring the computer science curriculum is a complex but fundamental process for ensuring relevant education. The integration of emerging technologies, adjustment of content, involvement of the educational community, and impact assessment are key elements in developing a flexible curriculum aligned with the needs of contemporary society. Through careful monitoring and continuous adaptation of the curriculum, schools can provide students with fundamental training, facilitating their success in a digitalized and dynamic society. Although significant progress has been made in structuring and developing the curriculum, there are still significant challenges that may hinder its complete and effective implementation.

The main challenges identified include insufficient technological infrastructure, limited hours allocated to the subject, uneven teacher preparation, and the lack of comprehensive and interactive educational resources. These obstacles require integrated solutions, such as revising the allocation of computer science hours, investing in equipment and technologies, continuous teacher training, and the development of supplementary educational resources.

Furthermore, collaboration between authorities, teachers, parents, and the community is essential to create an appropriate educational environment and ensure quality digital education. Investments in digital education not only improve students' preparation for the future but also contribute to the progress of a knowledge-based and innovative economy. Thus, through a strategic and sustainable approach, the necessary skills can be developed to ensure that young people are prepared for the challenges and opportunities of an increasingly technologized world.

References:

1. BĂLȚĂTESCU, S. (2008). *Managementul educaţional şi inovarea didactică.* Editura Didactică şi Pedagogică
2. *Curriculumul național pentru învățământul gimnazial. Disciplina Informatica* (2019). Ministerul Educației, Culturii și Cercetării al Republicii Moldova. Chișinău. https://mecc.gov.md/sites/default/files/informatica_gimnaziu_2019-2020_final.pdf
3. *Scratch programming, Medium.* https://medium.com/@shangayu8/scratch-programming-3cae3251a36f
4. *Kahoot* site oficial https://kahoot.com/

Note: *This article was developed as part of the project „Theory and methodology of continuous and cyclical monitoring and development of the school curriculum”, code 01.14.0, coordinated by the Center for Scientific Research „Educational and Social Policies”.*

Information about the authors:
Maria CRISTEI, doctor, associate professor, Faculty of Mathematics and Computer Science, State University of Moldova.
E-mail: cristeimusm@yahoo.com
ORCID: 0000-0002-8537-8728

Ghenadie MARIN, university lecturer, Faculty of Mathematics and Computer Science, State University of Moldova.
E-mail: ghena.marin@yahoo.com
ORCID: 0009-0000-5073-692X

Lilia IVANOV, Teacher, second degree teacher, Ion Creangă Theoretical High School, Coșnița village, Dubăsari district.
E-mail: ivanov.lilia1976@gmail.com
ORCID: 0009-0001-8735-5300

Prezented on 26.09.2024