International Journal on Integrated Education (IJIE)

e-ISSN: 2620 – 3502
p-ISSN: 2615 – 3785

Volume: 9 Issue: 4 | December 2026

https://journals.researchparks.org/index.php/IJIE

*Article*

# Pedagogical and Methodological Advantages of Implementing Digital and Intelligent Diagnostic Environments in Higher Education

**Gulnur Sagibaevna Tleubaeva**[1]

1. Independent Researcher, Bukhara State Technical University, Bukhara, Uzbekistan

**Abstract:** This article theoretically analyzes the pedagogical and didactic advantages of implementing intelligent diagnostic digital learning environments designed to identify and develop students' programming aptitude in the context of higher education digitalization and the credit-module system. Using analysis of the scientific literature and regulatory documents, comparative analysis and pedagogical modeling, the study describes mechanisms for constructing personalized learning pathways based on the principles of asynchronous and differentiated learning, and for organizing process-based assessment with learning analytics and a Socratic artificial intelligence (AI) tutor. A comparative analysis of traditional learning environments and intelligent diagnostic environments is presented, together with a three-stage methodology for instructional design that integrates diagnostic profiling, the PRIMM (Predict–Run–Investigate–Modify–Make) teaching cycle and process-based final assessment. The proposed framework is conceptual; its effectiveness should be verified through empirical studies in programming courses.

**Keywords:** Learning Analytics, Socratic AI Tutor, Differentiated Learning, Process-Based Assessment, Credit-Module System

## 1. Introduction

Digitalizing educational processes, organizing students' independent learning in line with the standards of the credit-module system and training competitive IT specialists for the digital economy are among the priority tasks set by the "Digital Uzbekistan – 2030" Strategy [1]. The Concept for the Development of the Higher Education System of the Republic of Uzbekistan until 2030 likewise identifies the digital transformation of teaching and the expansion of independent learning as key directions of reform [2]. In cognitively demanding fields such as programming and computer science, students differ substantially in baseline knowledge, speed of logical reasoning and individual learning pace.

In traditional educational environments, instructors are compelled to deliver a single standardized curriculum and assignments of uniform complexity to all students, oriented toward an average level of preparation. On the one hand, this artificially constrains the cognitive growth of high-achieving students; on the other, it imposes excessive cognitive load on underprepared students, which leads to frustration, academic failure and dropout [3].

To resolve this methodological contradiction, higher education institutions can implement intelligent diagnostic digital learning environments that profile students'

**Citation:** Tleubaeva, G. S. Pedagogical and Methodological Advantages of Implementing Digital and Intelligent Diagnostic Environments in Higher Education. International Journal on Integrated Education (IJIE) 2026, 9(4), 85-90.

Received: 10th Aug 2026
Revised: 30th Aug 2026
Accepted: 20th Sep 2026
Published: 22nd Oct 2026

**Copyright:** © 2026 by the authors. Submitted for open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/)

individual cognitive capabilities and provide personalized learning pathways. The aim of this study is to substantiate the pedagogical and methodological advantages of such environments for programming education and to propose a stage-based methodology for instructional design.

The pedagogical, didactic and conceptual foundations of developing digital learning resources and digital learning environments were laid by the Uzbek scholar U. Sh. Begimqulov, who showed that the implementation of digital packages in education requires adherence to the principles of interactivity, visual representation, orientation toward independent knowledge acquisition and motivational impact [4]. N. I. Taylaqov extended these foundations to the design and implementation of educational and methodological complexes for distance learning [5].

In the context of higher education digitalization and the credit-module system, B. Sh. Usmonov analyzed the main stages and directions of digital education development in Uzbekistan [6]. M. Abdurashidova and co-authors demonstrated the effect of innovation and digitalization on the quality of higher education at selected Uzbek universities [7]. G. Xolmirzayeva showed that digital platforms such as Moodle and HEMIS make it possible to organize independent study hours efficiently and to monitor academic engagement remotely [8].

At the international level, S. R. Nurdin and co-authors developed an application for testing the programming ability of informatics students on the basis of randomized algorithmic tests [9]. Within J. Sweller's cognitive load theory, worked examples and gradual task complexity reduce the extraneous load on novice learners. R. Lister and colleagues used the SOLO taxonomy to show that many novice programmers can read individual lines of code but fail to grasp the purpose of a program as a whole [10]. S. Sentance, J. Waite and M. Kallia proposed the PRIMM cycle as a structured approach to teaching programming through reading and modifying code before writing it [11].

In the era of generative AI and learning analytics, A. Atta and co-authors demonstrated that a ChatGPT-based mobile application can improve students' C# programming skills [12]. M. Jukiewicz systematically compared large language models as tools for the automated assessment of programming assignments [13]. E. Frankford and colleagues reported on the integration of an AI tutor into programming assessments and emphasized the need for pedagogical control over the hints it provides [14]. However, these studies do not offer an integrated didactic model that combines diagnostic profiling, differentiated learning pathways and process-based assessment within the credit-module system, which defines the research gap addressed in this article.

## 2. Materials and Methods

The study is theoretical and methodological in nature. It is based on an analysis of the scientific literature on digital learning environments, programming education and AI-supported assessment, as well as on the regulatory documents of the Republic of Uzbekistan on the digitalization of higher education. The following methods were used: theoretical analysis and synthesis, comparative analysis of traditional and intelligent diagnostic learning environments, and pedagogical modeling of the instructional design process. The study also draws on the author's earlier work on the development and application of AI in education [15].

The comparison of learning environments was carried out according to six criteria: the object of evaluation, instructional pacing, feedback, the nature of tasks, the degree of subjectivity of assessment and the role of AI. No empirical experiment was conducted at this stage; the results are therefore presented as a substantiated conceptual model.

## 3. Results

### Pedagogical Advantages of Intelligent Diagnostic Environments

An intelligent diagnostic digital learning environment for programming education differs from traditional teaching packages and static digital textbooks in five fundamental pedagogical respects.

*Asynchronous and personalized learning.* Students work with learning materials independently of a fixed class schedule, at a convenient time and place and at their own pace. Auto-grader and automated testing modules embedded in the system evaluate student code in real time (24/7) and provide immediate feedback without direct intervention by the instructor.

*Adaptive and differentiated approach.* Based on their baseline logical and algorithmic preparation, students are assigned to three groups, in line with the author's model for identifying talented students with AI algorithms [16]:

- High level (analytical/creative): the continuous development of high-achieving students is supported through complex, non-standard tasks, Parsons puzzles and project-based assignments.
- Intermediate level (technological): standard algorithmic and programming problem-solving tasks.
- Low level (executive/at-risk): scaffolding and worked examples are provided to reduce cognitive load.

Parsons puzzles, in which students arrange given code fragments in the correct order, are particularly suitable for differentiation because they isolate algorithmic logic from syntax [17].

*Socratic AI tutor and cognitive support.* The core didactic rule of the integrated AI module is never to provide ready-made code. The AI tutor applies the Socratic method: it asks guiding questions and prompts students to locate logical errors on their own. This discourages copying ready-made solutions and encourages independent logical and metacognitive reasoning.

*Process-based assessment and academic integrity.* Whereas traditional education evaluates only the final code submission, an intelligent diagnostic environment assesses the digital trajectory of code creation. The system analyzes students' digital footprints: time on task, the number of compilation attempts, the code editing history (keystroke dynamics) and the prompts submitted to the AI (prompt logs). This approach substantially reduces the risk of plagiarism and uncritical code copying.

*Learning analytics and monitoring for instructors.* Through a dedicated dashboard, instructors monitor the progress of the whole group and of individual students in real time. This supports evidence-based lesson planning and targeted intervention for struggling students.

### Stage-Based Methodology for Instructional Design

Instruction in an intelligent diagnostic digital environment is organized in three sequential pedagogical stages (Table 1).

**Table 1.** Stages of instructional design in an intelligent diagnostic environment

| Stage | Pedagogical content |
|---|---|
| Stage 1. Introductory (pre-test) and profiling | • Assessment of the student's baseline logical-algorithmic potential<br>• Syntax-free logical and algebraic diagnostic tests |
| Stage 2. Formative (active-diagnostic) | • PRIMM teaching cycle |

| | |
|---|---|
| | • Practice with the Monaco code editor and the Socratic AI tutor<br>• Cognitive load management (worked examples and Parsons puzzles)<br>• Functional code testing with an auto-grader (Judge0 API) |
| Stage 3. Final monitoring and dialogic defense (post-test) | • Evaluation of process-based indicators (PASI, CEI, EEI)<br>• Oral verification (code review) and determination of the individual level |

**Comparison of Traditional and Intelligent Diagnostic Environments**

Table 2 compares the traditional learning environment with the intelligent diagnostic environment according to the six criteria described in the methodology.

**Table 2.** Comparative analysis of traditional and intelligent diagnostic learning environments

| Criterion | Traditional learning environment | Intelligent diagnostic environment |
|---|---|---|
| Object of evaluation | Only the final code file or written assignment | Digital trajectory of code creation and the cognitive process (process-based) |
| Instructional pacing | Uniform for all students (average group pace) | Personalized, asynchronous and adapted to individual cognitive capabilities |
| Feedback | Delayed (several days after manual grading) | Immediate automated feedback (auto-grader and AI tutor) |
| Nature of tasks | Uniform, static exercises and tests | Differentiated and adaptive tasks, including Parsons puzzles |
| Degree of subjectivity | Risk of subjectivity due to the human factor | Reduced subjectivity through learning analytics and digital indicators |
| Role of AI | Restricted or banned (uncontrolled use) | Integrated under pedagogical guidance as a Socratic AI tutor |

## 4. Discussion

The analysis indicates that the main pedagogical value of intelligent diagnostic environments lies not in automation as such, but in shifting the focus of assessment from the final product to the learning process. This shift makes it possible to identify students who need support earlier and to offer high-achieving students tasks that match their potential. The proposed three-stage methodology connects these possibilities with established pedagogical approaches: diagnostic profiling, the PRIMM cycle and cognitive load management.

At the same time, several limitations must be taken into account. Automated indicators reduce, but do not eliminate, subjectivity, since the choice of indicators and their thresholds remains a pedagogical decision. Recording keystroke dynamics and prompt logs requires transparent rules on personal data protection and informed student consent. The quality of AI-generated hints depends on the underlying language model and must be monitored by the instructor. Finally, the advantages described in this article are

substantiated theoretically; their effect on learning outcomes should be tested in controlled pedagogical experiments.

## 5. Conclusion

Implementing intelligent diagnostic digital environments to identify and develop students' programming aptitude is an important factor in improving the quality of higher education. Such an environment overcomes the traditional constraint of "teaching to the average" by offering each student a personalized learning pathway that corresponds to his or her cognitive potential. On this basis, the following practical recommendations are proposed:

1. Redesign programming curricula and lesson scenarios in higher education institutions on the basis of the PRIMM cycle and the Socratic AI tutor principle.
2. Introduce a multi-factor rating system that relies not only on final grades but also on students' digital footprints (learning analytics) and process-based performance indicators.
3. Conduct empirical studies to validate the proposed process-based indicators and to evaluate the effect of the environment on learning outcomes.

## 6. Acknowledgement

The author thanks the faculty of Bukhara State Technical University for their support. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## REFERENCES

[1] *Decree of the President of the Republic of Uzbekistan "On Approval of the 'Digital Uzbekistan – 2030' Strategy and Measures for Its Effective Implementation,"* No. UP-6079, Oct. 5, 2020. [Online]. Available: https://lex.uz/docs/5031048

[2] *Decree of the President of the Republic of Uzbekistan "On Approval of the Concept for the Development of the Higher Education System of the Republic of Uzbekistan until 2030,"* No. UP-5847, Oct. 8, 2019. [Online]. Available: https://lex.uz/docs/4545887

[3] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cogn. Sci.*, vol. 12, no. 2, pp. 257–285, 1988, doi: 10.1207/s15516709cog1202_4.

[4] U. Sh. Begimqulov, *Pedagogical Sciences: Theory and Methodology of E-Learning*. Tashkent, Uzbekistan: Fan, 2005.

[5] N. I. Taylaqov, *Creation and Implementation of Distance Learning Educational and Methodological Complexes*. Tashkent, Uzbekistan: O'qituvchi, 2005.

[6] B. Sh. Usmonov and U. B. Shukurillaev, "Brief overview of digital education development in Uzbekistan," *Human Capital*, vol. 2, no. 12(168), pp. 13–19, 2022.

[7] M. Abdurashidova, M. E. Balbaa, S. Nematov, Z. Mukhiddinov, and I. Nasriddinov, "The impact of innovation and digitalization on the quality of higher education: A study of selected universities in Uzbekistan," *J. Intell. Syst.*, vol. 32, no. 1, Art. no. 20230070, 2023, doi: 10.1515/jisys-2023-0070.

[8] G. Xolmirzayeva, "The role and importance of applying digital technologies in higher education system today," *Green Economy and Development*, vol. 3, no. 9, 2025, doi: 10.5281/zenodo.17161444.

[9] S. R. Nurdin et al., "Development of programming ability test application for informatics students," *J. Phys.: Conf. Ser.*, vol. 1569, Art. no. 032098, 2020, doi: 10.1088/1742-6596/1569/3/032098.

[10] R. Lister, B. Simon, E. Thompson, J. L. Whalley, and C. Prasad, "Not seeing the forest for the trees: Novice programmers and the SOLO taxonomy," in *Proc. 11th Annu. SIGCSE Conf. Innov. Technol. Comput. Sci. Educ. (ITiCSE)*, Bologna, Italy, 2006, pp. 118–122, doi: 10.1145/1140124.1140157.

[11] S. Sentance, J. Waite, and M. Kallia, "Teaching computer programming with PRIMM: A sociocultural perspective," *Comput. Sci. Educ.*, vol. 29, no. 2–3, pp. 136–176, 2019, doi: 10.1080/08993408.2019.1608781.

[12] A. Atta, M. Esmat, N. Amasha, E. Elayat, and W. K. ElSaid, "A smart ChatGPT mobile application for improving C# programming skills for students in educational institutions," *Qubahan Acad. J.*, vol. 5, no. 2, pp. 49–62, 2025, doi: 10.48161/qaj.v5n2a1772.

[13] M. Jukiewicz, "A systematic comparison of large language models for automated assignment assessment in programming education: Exploring the importance of architecture and vendor," arXiv:2509.26483, 2025.

[14] E. Frankford, C. Sauerwein, P. Bassner, S. Krusche, and R. Breu, "AI-tutoring in software engineering education: Experiences with large language models in programming assessments," in *Proc. 46th Int. Conf. Softw. Eng.: Softw. Eng. Educ. Training (ICSE-SEET)*, Lisbon, Portugal, 2024, pp. 309–319, doi: 10.1145/3639474.3640061.

[15] G. Tleubaeva, "The development and application of artificial intelligence in education," *Science and Education in Karakalpakstan*, no. 4/2, pp. 347–351, 2025. [Online]. Available: https://cdn.karsu.uz/uploads/qft-jurnal/e2ffde3c-a534-4ad2-8956-dcbe6920d324.pdf

[16] G. Tleubaeva, "Sun'iy intellekt algoritmlari yordamida iqtidorli talabalarni tanlash modelini ishlab chiqish [Development of a model for selecting talented students using artificial intelligence algorithms]," *Muǵallim hám úzliksiz bilimlendiriw*, no. 5/2, pp. 277–282, 2025. [Online]. Available: https://mugallim-journalkk.uz/wp-content/uploads/2026/07/5-2-2025.pdf

[17] D. Parsons and P. Haden, "Parson's programming puzzles: A fun and effective learning tool for first programming courses," in *Proc. 8th Australasian Comput. Educ. Conf. (ACE)*, CRPIT vol. 52, Hobart, Australia, 2006, pp. 157–163.