AMERICAN JOURNALS
Open Access, Peer Reviewed Journals

American Journal of Pedagogical and Educational Research
ISSN (E): 2832-9791| Volume 47, |April - 2026

REQUIREMENTS FOR A SOFTWARE MODEL THAT DEVELOPS STUDENTS' INDEPENDENT LEARNING COMPETENCE

Tursunova Zulayxo Omonulayevna
Teacher of the Department of Mathematics at National
Pedagogical University of Uzbekistan Named after Nizami
E-mail: zulaykho.tursunova.86@mail.ru

A B S T R A C T

This article provides a scientific and theoretical analysis of the requirements for a software model aimed at developing students' independent learning competence. The increasing importance of independent learning in the modern education system is substantiated, and issues related to the effective organization of students' activities through a digital learning environment are discussed. The article identifies constructivism, the competency-based approach, and adaptive learning principles as the pedagogical foundations of the improved software model. In addition, the functional capabilities of the model are clarified, including personalization, self-monitoring, prompt feedback, and requirements for an Uzbek-language interface. From a technical perspective, important aspects such as web architecture, mobile adaptability, database systems, and information security are examined. As a result, an integrated software environment model is proposed that supports the stages of independent learning, including motivation, planning, implementation, assessment, and reflection.

K E Y W O R D S

Independent learning competence, software model, digital learning environment, constructivism, competency-based approach, adaptive learning, personalization, self-monitoring, feedback, reflection, credit-module system, higher education.

INTRODUCTION

The rapid development of science and technology, the sharp increase in information flow, and the changing demands of the labor market impose new tasks on the education system. A modern specialist must not only possess a certain amount of theoretical knowledge, but also be able to independently acquire new knowledge, analyze problems, and effectively organize their activities. Therefore, developing students' independent learning competence is considered one of the priority tasks of the higher education system today [1].

In the modern education system, the formation of independent learning competence largely depends on how the educational process is organized, as well as on teaching methods, educational technologies, and the level of student engagement in learning activities. In particular, the digital learning environment formed on the basis of digital technologies significantly expands students' opportunities for independent learning. It enables students to quickly access educational materials, use various electronic resources, complete assignments remotely, and independently monitor their knowledge.

Studies show that the use of electronic learning environments plays an important role in developing students' independent learning competence. However, to organize this process effectively, specialized pedagogical methodologies and appropriate software support are required.

The development of an improved software model aimed at enhancing students' independent learning competence is one of the important scientific and practical tasks of the modern higher education system. The relevance of this task is primarily explained by the fact that, under the credit-module system, a significant part of student activity is carried out through independent learning outside the classroom. Therefore, independent work should not be treated as a simple set of assignments, but rather as a goal-oriented, managed, monitored, and assessed pedagogical process. The effective organization of such a process requires a well-designed software model.

The improved software model should be pedagogically grounded, functionally rich, and technically stable. It should not serve merely as a tool for storing educational materials or collecting assignments, but rather function as an integrated digital environment that supports students' independent learning activities step by step, takes into account their individual needs, analyzes learning outcomes, and ensures interaction with the teacher. Therefore, it is appropriate to classify the requirements for such a model into three main groups: pedagogical, functional, and technical requirements.

The effectiveness of any educational software system is primarily determined by the pedagogical theory on which it is based. If a software tool is not aligned with the internal principles of education, the psychological and pedagogical mechanisms of student activity, and modern teaching principles, it cannot ensure effective learning outcomes, no matter how technically advanced it may be. From this perspective, the improved software model should rely on three key methodological foundations: constructivism, the competency-based approach, and adaptive learning principles.

According to the constructivist approach, knowledge is not transmitted in a ready-made form from teacher to student; rather, it is constructed through the student's own activity, experience, problem-solving, observation, and reflection. In other words, the student is not a passive recipient of knowledge, but an active constructor of it. Therefore, the improved software model should support this active role of the student.

In such a model, educational content should not be limited to simple texts, lectures, or test questions. Instead, it should be presented in a way that stimulates students' thinking, encourages them to make choices, analyze, compare different sources, and draw independent conclusions. For example, the model should include problem-based tasks, case studies, project assignments, inquiry-oriented questions, open-ended tasks, reflective blocks, and discussion forums.

From the perspective of constructivism, the software system itself should function as a "learning environment." That is, it should not merely be a repository of content, but a space where students independently construct knowledge, learn from their mistakes, make repeated attempts, receive feedback, and analyze their results. This requires the system to include the following pedagogical features:

- the gradual increase in the complexity of learning tasks;
- content that builds on the student's prior knowledge and experience;
- instruction based on problem situations;
- tasks oriented toward independent inquiry;
- elements of collaborative activity;
- assessment that takes into account not only the outcome but also the learning process.

Thus, from the standpoint of constructivist pedagogy, the improved software model should support the student as an active, independent, and conscious subject of learning rather than a passive participant.

The primary goal of modern higher education is not only to provide a certain amount of theoretical knowledge, but also to prepare competent specialists who can apply this knowledge in real educational, professional, and social contexts. Therefore, the improved software model should also be developed based on the competency-based approach.

Independent learning competence refers to a student's ability to independently set goals, plan tasks, select necessary resources, complete assignments, monitor their own activities, evaluate results, and adjust their subsequent actions. The software model should be specifically designed to develop these components.

The competency-based approach imposes the following requirements on the model:

- the system should encourage students to make independent decisions;
- tasks should be aligned with real-life and professional situations;
- outcomes should be assessed not only through theoretical knowledge but also through practical activities;
- modules for self-regulation, self-assessment, and reflection should be included;
- the student should be considered a responsible subject for their own learning outcomes;
- individual learning trajectories and personal development dynamics should be monitored throughout the learning process.

It is important to emphasize that the competency-based approach requires not merely the completion of tasks, but the organization of activities. Therefore, the system should not be limited to instructing students on what to do; it should also explain why a task is important, how it can be effectively completed, how the results should be evaluated, and what conclusions should be drawn from them.

Higher education practice shows that not all students learn with the same level of preparation, pace, needs, or motivation. Under such conditions, a platform that offers the same tasks, pace, and pathway for everyone cannot be sufficiently effective. Therefore, the improved model should be built on the principles of adaptive learning. Adaptive learning implies that the system collects data about the user, analyzes their learning activities, and, based on this, provides individualized recommendations, tasks, and learning trajectories. This approach is especially important in independent learning, as students often work without direct teacher supervision, and the system itself should act as a kind of digital assistant.

The principles of adaptive learning should be reflected in the model in the following forms:

- determining the student's level through initial diagnostics;
- adjusting the complexity of tasks based on performance;
- providing additional materials for weak topics;
- offering more complex and in-depth tasks for high-performing students;
- giving individual recommendations and alerts;
- allowing the selection of content in various formats.

Thus, the model should aim to create not a "one-size-fits-all course," but a personalized learning path for each student.

One of the most important features of the improved software model is that it should view the independent learning process as an integrated, continuous, and manageable system. Independent work

is often understood merely as "assigning tasks and submitting results." In reality, however, this process includes at least five stages: motivation, planning, implementation, assessment, and reflection. The software model must support all of these stages.

The starting point of any independent activity is motivation. If a student does not clearly understand why a particular task needs to be completed, what results are expected, and how it will contribute to their academic and professional development, the task is often performed formally. Therefore, the model should include motivational components.

At this stage, the system should:

- clearly define the purpose of the task;
- demonstrate the practical and professional relevance of the topic;
- present the expected outcomes in a visual form;
- stimulate interest through gamification elements;
- provide opportunities to compare personal results with previous performance;
- outline a step-by-step path to achieving success.

A system that supports the motivation stage activates the student's intrinsic interest and facilitates engagement in independent learning.

After motivation, the student should be able to plan their activities. At this stage, decisions are made regarding when and in what sequence tasks should be completed, and which resources should be used. The software model should support this stage through dedicated planning tools.

For example, the system should include:

- a calendar and deadline system;
- "to-do" lists;
- modules that break tasks into smaller steps;
- indicators showing time allocation;
- options for setting task priorities;
- reminders and notifications.

These features help develop students' skills in time management, proper workload distribution, and proactive organization of their activities.

The implementation stage is the central part of independent learning, where the student directly engages in learning activities. At this stage, the system should provide the greatest practical support. To support the implementation process, the model should include the following:

- content in various formats: text, video, audio, infographics, and interactive materials;
- step-by-step instructions for completing tasks;
- additional explanation and guidance modules;
- examples and sample solutions;
- guiding questions;
- automatic saving and continuation options;
- tracking of user activity.

At this stage, it is important for the system to act as a "digital tutor." It should not only assign tasks, but also guide and support the student throughout the execution process.

The next stage of independent learning is assessment. The purpose of assessment is not only to record the final result but also to ensure educational development. Therefore, the system should organize assessment in a transparent, clear, multi-criteria, and development-oriented manner.

The assessment module should include:
- automated test evaluation;
- rubrics for open-ended tasks;
- formative (intermediate) assessment;
- summative (final) assessment;
- self-assessment;
- peer assessment elements;
- detailed presentation of results based on specific criteria.

The assessment system should be understandable for the student: they should clearly see the criteria on which they are evaluated, as well as their strengths and weaknesses.

One of the highest stages of independent learning competence is reflection. If a student does not analyze their own activities, the process of independent learning becomes repetitive and mechanical. Therefore, the system should also include reflection modules.

This module should include:
- reflective questions;
- “What have I learned?” windows;
- error analysis;
- recommendations for repeated attempts;
- graphs of personal development dynamics;
- recommendations for next steps.

Reflection helps students consciously manage their own activities and transforms independent learning competence into a long-term developmental process.

Based on pedagogical foundations, the functional capabilities of the model should also be clearly defined. They determine how the system operates from the user’s perspective, what services it provides, and how it supports the educational process.

The model should take into account the individual characteristics of each student. Personalization should include the following:
- creating an individual profile;
- tasks adapted to the student’s level of preparedness;
- recommendations based on learning activity;
- the possibility to choose appropriate content formats;
- an individual learning trajectory.

Without personalization, the model will remain at the level of a standard LMS. The essence of the improvement proposed in this research should be reflected precisely in this function.

Students should be able to independently monitor their own activities. For this purpose, the system should include:
- a progress panel;
- a list of completed and pending tasks;
- learning performance dynamics;
- time expenditure indicators;
- indicators signaling weak topics.

These tools help students manage their learning activities and strengthen their sense of responsibility.

To make independent learning effective, prompt feedback is essential. The system should provide the following types of feedback:
- immediate display of results;
- explanations of errors;
- recommendations for the next steps;
- advice tailored to individual performance;
- motivational support messages.

Explanatory feedback has greater educational value than simple “correct” or “incorrect” responses.
In the context of national higher education, it is particularly important for the model to function fully in the Uzbek language. This includes:
- menus, buttons, and notifications in Uzbek;
- correct use of scientific and methodological terminology;
- high-quality preparation of tasks and content in Uzbek;
- the availability of search and recommendation features in Uzbek.

This requirement is important not only for user convenience but also for the development of national educational content.
If pedagogical and functional requirements are not supported by a technically robust platform, the system cannot operate reliably in practice. Therefore, the technical requirements of the model must also be clearly defined.
The model should be built on a modern web architecture. This implies:
- operation through a web browser;
- a client-server architecture;
- a modular structure;
- scalability and ease of extension;
- integration with other systems via APIs.

A modular architecture facilitates the addition of new functionalities in the future, such as artificial intelligence modules or analytics components.
Considering that a large proportion of students access the platform via mobile devices, the system must be compatible with mobile devices. This includes:
- responsive design;
- full functionality in mobile browsers;
- development of a mobile application if necessary;
- the ability to function even with low internet speed.

The model must include a reliable database to store users, tasks, results, analytics, and content. It should be:
- stable;
- scalable;
- fast;
- capable of backup;
- suitable for analytical queries.

In addition, to support artificial intelligence or recommendation modules, the database should be capable of storing user activity history.

Educational systems handle personal and academic data; therefore, special attention must be given to security. The model should include:
- authentication;
- authorization;
- data encryption;
- backup mechanisms;
- restriction of unauthorized access;
- logging and auditing systems.

Thus, the improved software model should not be a simple electronic platform, but a well-founded educational environment in pedagogical, functional, and technical terms. Its pedagogical foundation is based on constructivism, the competency-based approach, and adaptive learning principles. The model must fully support all stages of the independent learning process, including motivation, planning, implementation, assessment, and reflection. Personalization, self-monitoring, automated feedback, and an Uzbek-language interface are essential functional requirements of the model. Meanwhile, modern web architecture, mobile adaptability, a reliable database, and security ensure its technical stability. Only a model developed through such a multi-layered approach can become an effective software tool that genuinely enhances students' independent learning competence.

References

1. Knowles M. Self-Directed Learning: A Guide for Learners and Teachers. New York: Cambridge Books, 1975.
2. Zimmerman B.J. Self-Regulated Learning and Academic Achievement: An Overview. Educational Psychologist, 2002.
3. Pintrich P. The Role of Goal Orientation in Self-Regulated Learning. Handbook of Self-Regulation, 2000.
4. Boekaerts M. Self-Regulated Learning: Where We Are Today. International Journal of Educational Research, 1999.
5. Garrison D.R. Self-Directed Learning: Toward a Comprehensive Model. Adult Education Quarterly, 1997.
6. Azevedo R., Cromley J. Does Training on Self-Regulated Learning Facilitate Students Learning With Hypermedia? Journal of Educational Psychology, 2004.
7. Greene J. A., Azevedo R. Learning With Computer-Based Learning Environments. Review of Educational Research, 2010.
8. Anthonysamy L. Self-Regulated Learning Strategies and Non-Academic Outcomes in Higher Education. International Journal of Educational Technology in Higher Education, 2020.
9. Al-Adwan A., Al-Adwan A. E-learning Readiness and Self-Directed Learning in Higher Education. Education and Information Technologies, 2018.
10. OECD Definition and Selection of Key Competencies (DeSeCo Project). Paris: OECD Publishing, 2005.
11. Bates T. Teaching in a Digital Age: Guidelines for Designing Teaching and Learning. Vancouver: Tony Bates Associates, 2015.
12. Siemens G. Connectivism: A Learning Theory for the Digital Age. International Journal of Instructional Technology, 2005.

13. Anderson T. The Theory and Practice of Online Learning. Athabasca University Press, 2008.
14. Laurillard D. Teaching as a Design Science. Routledge, 2012.
15. Clark R., Mayer R. E-learning and the Science of Instruction. Wiley, 2016.
16. Зимняя И.А. Ключевые компетенции как результат образования. Москва, 2004.
17. Хуторской А.В. Ключевые компетенции и образовательные стандарты. Педагогика, 2002.
18. Полат Е.С. Педагогические технологии дистанционного обучения. Москва, 2006.
19. Кутепова Л.И. Организация самостоятельной работы студентов в электронной среде. Москва, 2017.
20. Шурыгин В.Ю. Использование Moodle для организации самостоятельной работы студентов. Казань, 2018.
21. Ляшенко М.С. Использование Moodle при обучении иностранным языкам. Москва, 2019.
22. Азжеурова М.В. Электронная информационно-образовательная среда в вузе. Санкт-Петербург, 2016.
23. Begimqulov U.Sh. Scientific and Theoretical Foundations of the Introduction of Information Technologies in Pedagogical Education. Tashkent, 2007.
24. Ishmukhamedov R.J. Innovative Pedagogical Technologies. Tashkent, 2010.
25. Tolipov O‘.Q., Usmonboev M. Theory and Practice of Pedagogical Technologies. Tashkent, 2012.
26. Muslimov N.A. Fundamentals of Pedagogical Competence. Tashkent, 2014.
27. Abduqodirov A.A. Information Technologies and Education. Tashkent, 2013.
28. Saidov S. Methodology for Organizing Independent Learning in Higher Education. Tashkent, 2018.
29. Qurbonova U. Didactic Foundations of Organizing Students’ Independent Work. Tashkent, 2019.
30. Concept for the Development of the Higher Education System of the Republic of Uzbekistan until 2030. Tashkent, 2019.