The Second International Scientific Conference on Economics, Management and Information Technologies – ICEMIT 2025
CONFERENCE PROCEEDINGS

Taxonomy development goals for educational software design

Srećko Stamenković[a*], Miloš Ilić[a], Bojan Vasović[a], Zoran Jovanović[a], Ivan Garvanov[b]

[a] Toplica Academy of Applied Studies, Department of Business Studies Blace, Serbia
[b] University of Library Studies and Information Technologies, Sofia, Bulgaria

Article info

Review paper

DOI:
https://doi.org/10.71159/icemit2564S

Abstract

In modern education, the development of educational software requires careful alignment of different goals in order to ensure an effective, pedagogically relevant and technically stable learning experience. This paper aims to explore and systematize taxonomic approaches applied in the design of educational software systems. Based on the analysis of relevant literature in the fields of pedagogy, software engineering and digital technologies in learning, a three-dimensional taxonomy of goals is formulated, which includes: (1) technical goals, (2) pedagogical goals, and (3) usability goals. This structure enables a holistic approach to the design of educational tools and provides a basis for evaluating and improving existing systems. The literature review indicates the fragmentation of existing taxonomies and the need to unify pedagogical and engineering principles into a single framework. The proposed taxonomy represents a step towards standardization and improvement of educational software design in accordance with modern educational needs.
Keywords: *taxonomy of goals, educational software, pedagogical goals, usability, technical requirements*

1. Introduction

In the last few decades, the development of educational software has made significant progress, driven by the growing demands of educational institutions, the digitalization of teaching and the individualized approach to learning. Although there is a wide range of available solutions, the quality and efficiency of educational software systems often vary, which indicates the need for clearly defined goals in their design. A key step towards improving the quality of educational software is the establishment of a taxonomy of goals – a structured classification system that helps organize the educational intentions and functionalities of software solutions.

The taxonomy of goals in educational software design is a theoretical framework that allows designers, teachers and researchers to systematically analyze and structure software functionalities in accordance with educational outcomes. This approach not only encompasses technical and pedagogical aspects, but also allows for a better mapping between software functions and specific educational needs (Valente et al., 2025).

The use of existing educational taxonomies, such as the well-known Bloom's Taxonomy, allows for a precise classification of goals according to cognitive complexity – from memorization and comprehension, to analysis, evaluation and creation. Many authors apply this taxonomy in the design of software learning environments, emphasizing how it can structure teaching activities according to the levels of cognitive development of students. However, some authors also point out its limitations in describing mathematical and abstract educational goals (Ormell, 1974).

In addition to the educational side, it is important to include the technical dimension – such as system architecture, usability and integration with institutional processes. For example, the study by Maulana et al. (2025) presents the design of an information system for managing a library collection within a university, which supports accreditation and resource availability via a web platform. This example shows how software designed according to clearly defined goals can simultaneously satisfy both educational and technical requirements.

*Corresponding author
E-mail address: s.stamenkovic@vpskp.edu.rs

On the other hand, modern educational environments require extended taxonomic models that include personalized learning dimensions, interactive functionalities and technical architectures of software systems. The aim of this paper is to explore, systematize and propose a taxonomy of goals that can serve as a foundation for the design of efficient, purposeful and pedagogically sound educational software systems. The taxonomy should have clear and precise goals and should be verified by an educational software system that is developed on the basis of the proposed taxonomy.

2. Literature review

Modern educational systems increasingly rely on software support to enhance personalization, content customization, and students' cognitive development. In this context, taxonomies of educational objectives and educational technology components play a key role in shaping teaching strategies, evaluation, and software tool development. This chapter aims to present a review of the relevant literature on taxonomies in education, with a special focus on their integration into educational software design and the definition of educational objectives.

A classic theoretical framework in educational practice is Bloom's Taxonomy, which structures educational objectives through a hierarchy of cognitive levels – from memorization to evaluation and creation. Ormell (1974) provides a critical review of Bloom's Taxonomy and questions its rigidity in relation to dynamic educational contexts. The author advocates a more flexible approach to goal formulation, emphasizing the need to link objectives to social and affective components of learning. He proposes an alternative approach based on imaginative understanding, which implies a flexible response to "if...then" questions.

The growing use of personalized learning systems has led to the need for new taxonomic classifications that go beyond traditional pedagogical frameworks. Ismail et al. (2023) offer a comprehensive taxonomy of personalized software systems, grouping them by type of environment (individual, collaborative), structure of learning content, and user models. This taxonomy allows for standardization in the development of educational software and better alignment with specific learning objectives. Nachmias et al. (1999) propose a functional taxonomy for the classification of educational websites, using criteria such as interaction type, purpose, and content format. Such a tool facilitates the analysis and design of web-based learning platforms.

In areas that combine IT and education, such as software engineering education, taxonomies serve to structure teaching units, tasks, and competencies. Pizard & Vallespir (2020) develop a taxonomy of educational outcomes in software engineering using empirical analysis of curricula and industry needs. Their taxonomy encompasses cognitive domains, but also professional skills such as teamwork, software lifecycles, and ethics. Usman et al. (2017) in their systematic study examine the methods of creating taxonomies in software engineering and propose a revised model that includes dimensions such as knowledge domain, target groups, and level of abstraction. Although the focus is not exclusively on education, the methodology for developing taxonomy can also be successfully applied to educational software.

Kagdi et al. (2007) present one of the most detailed taxonomies in the field of software mining (analysis of software repositories), applying it in the context of software evolution. The authors classify research into four dimensions: type of repository (what), purpose of analysis (why), methodology used (how), and evaluation methods (quality). This structure is highly transferable to educational software systems, as it allows for the analysis of iterative development and adjustment of software goals in accordance with teaching requirements. The literature confirms that taxonomies, whether pedagogical, functional, or engineering, play a key role in structuring educational goals and designing educational software systems. The integration of multiple taxonomic perspectives into a single framework can contribute to the development of higher-quality, scalable, and goal-oriented educational solutions. It is precisely such a framework that fills the current gap, as confirmed by a detailed literature review. The proposed taxonomy should be a framework that emphasizes an approach to the design of an educational system guided by clearly defined technical, pedagogical, and usability goals.

3. Identification and categorization of goals for educational systems design

The introduction and adoption of new information technologies in learning and teaching has evolved rapidly in recent years. The role of technology in higher education is to encourage students to think about the problem of study and to improve the educational process, and not to reduce it to a set of procedures for delivering content. Therefore, this paper lists the important aspects of software systems that are necessary for a system to be characterized as educational, or as a learning assistance system.

The research that involved finding an efficient solution for the development of an educational tool for learning program interpreters (Stamenković & Jovanović, 2023) has just influenced the formation of a universal taxonomy of goals for the development of educational tools. In order to clearly highlight the requirements for a new software system, a systematic study of the literature in the field of development and design of educational software systems for visualization of algorithms was carried out, and the problems encountered by such software when applied in the teaching process were also analyzed (Stamenković et al., 2020; Stamenković et al., 2023). Obstacles to the introduction of interactive dynamic visualizations in teaching include both pedagogical and usability issues (Amershi et al., 2008; Tversky, Morrison and Betrancourt, 2002). Therefore, before including any educational software in theoretical lectures or laboratory exercises, it is necessary to conduct evaluations of its effectiveness in terms of acquiring knowledge, understanding, engagement and motivation of students. Usability shortcomings relate to the time required to find and apply the appropriate functions of the tool, and then to the effort and time invested in learning how to use the tool (Naps et al., 2003).

Based on the extensive research conducted, it can be said that in the design phase, the main requirements for defining the design of an educational tool relate to the achievement of technical, pedagogical and usability goals. Within the framework of the taxonomy dimensions thus established, a classification of goals was also carried out, which can also be used for the evaluation of educational tools.

3.1. Technical goals

To define the architecture of a software system, it is necessary to consider all technical aspects that may have an impact on the final structure of all system components. The criteria that should be taken into account when designing the architecture are: platform, graphical environment, visualization control, knowledge assessment.

The paper (Stamenković & Jovanović, 2023) pointed out the potential problem of accessibility of systems that are implemented as desktop applications. Such systems are tied only to a specific operating system, and problems with the distribution of their installation packages often arise. Improving system functions or correcting identified deficiencies requires informing users, distributing a new version, and engaging users during installation. In order to overcome these problems, the educational system should be designed as a web application. In this way, independence from hardware and the operating system is ensured, and improving existing and adding new functions is done automatically and without user involvement. Educational content should not be presented only in textual form, but in an appropriate graphical environment whose appearance will ultimately be defined by pedagogical and usability goals. If possible, input data should not be entered only in text form, but through a graphical editor. The graphical user interface should be aesthetically pleasing with a minimalist design, so that the software is attractive, it should not display irrelevant or rarely used information. If the software performs a certain simulation or visualization of a process, it is necessary to provide a visualization control function. This means that when designing the system architecture, a component should be foreseen that will be responsible for extracting data on specific intermediate stages of the process that is being simulated or visualized. The possibility of testing student knowledge implies that the software should provide login to the system with user roles of student and teacher so that teachers have the option to create and save tests, and students have the option to solve tests and save data on results. In this way, a logged-in student would have access to their educational environment and their work results on any device and at any time. This implies the mandatory use of a database that should be foreseen in the architecture of the educational software system.

In the design phase, the basic technical requirement should be that the software system be easily upgradeable, due to possible future functions. Therefore, the educational tool should be designed as a modular software system that can have a set of submodules in the form of independent interactive tools.

3.2. Pedagogical goals

In order for a learning support software system to contribute to education, it must provide clear and specific pedagogical benefits. The following are the pedagogical goals, as well as the recommended features of the tools to achieve these goals, which are the basis for the design of educational software.

The theory of instruction that focuses on constructivist approaches points to the importance of active learning, that is, the active engagement of students in an educational environment that should encourage free interaction with information (Elissavet and Economides, 2000). Research has shown that students learn best when they are actively involved in the learning process itself (Chickering and Gamson, 1987). To achieve the effect of active learning, the educational environment should provide activities that are meaningful and that make students think about what they are doing (Bonwell and Eison, 1991). Schweitzer and Brown (2007) believe that in order to engage students, the educational environment should provide some of the following features:

- Interactivity (implies the existence of an activity that includes a feedback mechanism after the action taken by the student);
- Simplicity for understanding (if the activity requires a lot of explanation or is complex to understand, it distracts from the purpose and wastes valuable time);
- Short time frame (activities should be relatively short so that they complement the lectures, not replace them);
- Creativity (activities that are considered boring will not keep students interested and can be counterproductive);
- Collaboration (active learning activities can be effective as individual or group activities);
- Relevance (an irrelevant activity outside the context of the topic of study may break the boredom, but it does not contribute to the educational outcome of the lesson).

Nahvi (1996) believes that in order to motivate students, software systems must be intuitive, i.e. easy to use and understand. A typical way to motivate students is to inform them about what they will achieve at the end of the lesson by stating the objectives (Gagné, Briggs, Wager, 1988). In the context of an educational tool, before each step of the simulation/visualization, the student should be informed about what action is being taken and what it will achieve. Ensuring a measurable increase in understanding of the concepts and processes of the subject of study has traditionally been the focus of many researchers dealing with educational software systems (Hundhausen, 2002). Relevant content presented using a variety of graphical elements to help the student form accurate mental models of the algorithms can contribute to a deeper understanding of dynamic algorithms. One of the most important factors influencing differences in understanding is the students' prior knowledge (Adams et al., 1996). E-learning systems are expected to provide personalization, i.e. interaction with students taking into account their differences in knowledge levels, learning styles and priorities (Wade and Ashman, 2007). Therefore, when designing software, the fact that students may have different levels of prior knowledge should be taken into account. The tool should be adapted to both students with no prior knowledge and more advanced students, while supporting individual learning pace. In hypermedia learning systems, control is also important, which is primary in the design of interactive learning because it allows students to adapt the learning experience to their individual needs. However, there are dangers in handing over too much control to the user. Students with low abilities may become confused when control depends on a wide range of options (Litchfield, 1993). A high level of student control can lead to disorientation and distraction. However, control of the learning pace by the student is more appropriate than control that depends on the system. Innovative technology must give users a sense of control over the system. Students must be able to take a self-regulating role in the learning process so that they are aware that they are in control of their own learning (Ferdig, 2006). Hattie and Jaeger (1998) argue that assessment should be seen as an integral part of teaching and learning, not as an addition to it. It is necessary to create a student assessment system that takes into account educational goals and helps students develop their skills that will be useful in the long term (Ridgway, McCusker and Pead, 2004). Electronic assessment of knowledge improves the measurement of student outcomes and allows for immediate and direct feedback (Alruwais, Wills and Wald, 2018). After completing the study of a certain area or topic, the education system should provide the student with the opportunity to evaluate the acquired knowledge.

3.3. Usability goals

Many authors have provided different definitions of usability. In most cases, the common consensus is that a software product must first be functionally correct and then easy to use (Belson and Ho, 2012). Based on research from a theoretical and practical perspective in the field of software engineering, usability can be defined as the ease with which a user can learn the functioning of a system, that is, be able to prepare inputs and interpret the outputs of a system or component (Feizi and Wong, 2012). Ease of use is, however, a subjective matter. Usability as an ISO (International Organization for Standardization) quality criterion is explained in two current standards: ISO/IEC 9126-1: 2004 (understandability, learnability, operability and attractiveness) and ISO/IEC 9241-11: 2018 (effectiveness, efficiency and satisfaction), as well as in ISO/IEC 25010: 2011 (suitability, recognizability, operability, protection against user errors, user interface aesthetics and accessibility). From a software engineering perspective, usability refers to the user interface and the degree to which it satisfies various usability heuristics (Nielsen, 1993). The user interface should provide a simple dialogue between the system and the user that is clearly expressed in precise terms (Shneiderman and Plaisant, 2010). It should be consistent, and the need for the user to remember functions must be minimized (Norman and Draper, 1986). User errors are prevented by good design, but when error messages occur, the aforementioned dialogue guidelines are followed. Also, the system informs users of each step taken through appropriate feedback. Usability in educational technologies has been analyzed in various studies (Ardito, 2004; Storey et al., 2002; Hossain, 2015; Issa and Jusoh, 2019; Silius, Tervakari and Pohjolainen, 2003). These studies establish guidelines that guarantee the usability of educational technology (design and development) and criteria for verifying the level of usability of these educational systems.

There are numerous works that analyze the usability criteria of educational tools. Barker and King (1993) developed a method for evaluating interactive multimedia courses. They provide four factors that their research suggests are crucial for the usability of a software product (Table 1). They consider the design of the user interface to be a very important criterion, and the engagement of users in participatory tasks contributes to making the software system meaningful and thought-provoking.

Table 1. Essential usability criteria for interactive educational software

Category	Description
Quality of user interface design	Research shows that designers of top-rated products follow well-established rules and guidelines. This aspect of design affects the user's perception of the product (what they can do with it and how much it can engage them).
Engagement	Appropriate use of audio and moving visual segments can greatly contribute to the user's motivation to work with the medium.
Interactivity	Engaging in participatory tasks in a software product makes the user think.
Adaptability	Software products that allow users to configure and modify them to meet specific individual needs contribute to the quality of the educational experience.

Source: Barker & King (1993)

Nielsen and Mack (1994) defined a heuristic evaluation that addresses important usability issues. The criteria for their heuristic evaluation of educational software are shown in Table 2. The proposed criteria represent one of the popular definitions of usability. User control and free will are very important criteria, because the ability for users to leave the site and undo errors at any time provides an additional sense of satisfaction.

Table 2. Criteria for heuristic evaluation of educational software

Category	Description
System status display	The software informs the user about the status through appropriate and timely feedback.
Alignment of the system and the real world	The software uses the user's language, not jargon. Information appears in a natural and logical order.
Control and a sense of freedom	Users can leave sites and undo errors.
Consistency and standards	Users don't have to wonder if familiar commands, situations or actions in a given software mean the same thing. Common operating system standards are followed.
Error prevention	The design provides guidelines that reduce the risk of user error.
Recognition instead of memorization	Objects, actions and options are visible. The user does not have to rely on memory. Information is visible or easily accessible when needed.
Flexibility and efficiency of use	The software allows advanced users to use shortcuts and adjust settings as they see fit.
Aesthetic and minimalist design	The software provides an attractive overall design and does not display irrelevant or rarely used information.
Help with identifying, diagnosing and recovering from errors	Error messages are expressed in simple language, clearly indicate the problem and recommend a solution.
Help and documentation	The software provides appropriate help and documentation that is easy to access and related to the user's needs.

Source: Nielsen & Mack (1994)

Quinn (1996) believes that there is a possibility that educational software may not be usable or may have a high degree of usability, but the learning effect achieved is not satisfactory. Systems intended for learning must be both usable and educational, and on this basis, he proposed a heuristic evaluation method that assesses whether it is educational and usable software (table 3). The system must provide clear goals and tasks; the student should understand the actions he takes and know what he gets from using them.

Table 3. Heuristic evaluation method for usability of educational software

Category	Description
Clear goals and objectives	The student should understand what he can achieve by using the software.
Context with domain and student	The activity should be related to the exercise and should interest the student.
The content is clearly and multiple presented, with multiple navigation	The message should be unambiguous, support different student affinities, and allow the student to find relevant information while engaged in an activity.
Guided participation	Student support is needed to enable work within the competences.
Checking students' understanding	Students should articulate their conceptual understandings through appropriate feedback.
Formative evaluation	Students need constructive feedback on their progress.
The effect should be measurable.	Results should be clear and measurable.
Self-study support	The system should support the transfer of knowledge outside the intended learning environment and facilitate the student's self-improvement.

Source: Quinn (1996)

The usability of technology for teaching and learning also depends on the level of education for which it is intended, the characteristics of the students, and the specifics of the subject (Molina, Fuentes-Cancell and García-Hernández, 2022). Therefore, it is essential that educational software systems are aligned with literature and other resources in order to more effectively influence student understanding (Naps, Eagan and Norton, 2000; Kehoe, Stasko and Taylor, 2001). To achieve simple integration of tools, it must be easy for teachers to create new supporting materials or combine existing resources.

3.4. Taxonomy of goals

Based on an extensive analysis of the technical, pedagogical and usability objectives of educational software systems, a final taxonomy of objectives adapted to the design of tools suitable for learning assistance can be proposed. Table 4 shows the classification of explicit objectives that define the design of educational software systems. This table actually summarizes the design characteristics described by the objectives considered in this paper. This taxonomy contains only objectives that are universal and applicable to all learning assistance systems regardless of their domain of use. It helps in the design of new software systems, but is also effective for the evaluation of already existing tools. In this way, existing tools, before being used in a course, can be evaluated using this list and the tool that meets the highest percentage of the defined objectives can be selected.

Table 4. Taxonomy of objectives for the design of educational tools

Goal types	Goals
Technical	Independence from hardware and operating system Minimalist design The possibility of logging into the system Data security Easily upgradeable system
Pedagogical	Greater engagement Increasing motivation Deeper understanding of content Support for different levels of prior knowledge Individual pace of learning Assessment of knowledge
Usability	Easy to use Clear goals and objectives Control and free will Consistent user interface Feedback and help Error prevention Compliance with the literature

Source: Authors

4. Conclusion

The development of quality educational software requires clearly defined and mutually consistent objectives, which encompass multiple domains of knowledge and practice. Based on the literature review and our own research work, it is confirmed that a multidimensional taxonomy of objectives is crucial for the design of functional, pedagogically relevant and usable software solutions in education.

Literature from the fields of pedagogy (Ormell, 1974), personalized systems (Ismail et al., 2023), software engineering education (Pizard & Vallespir, 2020), as well as software systems taxonomy (Usman et al., 2017; Kagdi et al., 2007), indicates the need for an integrated approach in the classification of educational software objectives. These approaches clearly suggest that a one-sided perspective — either exclusively pedagogical or technical — is insufficient to respond to the complex demands of modern digital educational environments.

In this paper, the proposed taxonomy of objectives for educational software design encompasses three key domains:
- Technical objectives, which include system reliability, availability, interoperability, and data security;
- Pedagogical objectives, based on learning theories and didactic principles, including clearly defined educational outcomes and the development of cognitive skills;
- Usability objectives, which relate to interface intuitiveness, accessibility, navigation, and user experience.

Such a taxonomy enables a structured, evaluative and adaptive approach to the design of educational software systems. At the same time, it connects the theoretical basis (e.g., Bloom's taxonomy and competency models) with the practical aspects of engineering development and user interface, providing a clear framework for further development, testing and improvement of digital educational solutions. This work makes a significant contribution to the systematization of the process of developing educational software solutions, as it allows a multi-layered view of the requirements that educational systems need to meet in order to be functional, efficient and pedagogically relevant.

A multidisciplinary approach to the formation of a taxonomy of goals allows not only a better design of educational tools, but also lays the foundation for future standardization and comparison of different systems in educational practice and research. The next steps include empirical validation of the proposed taxonomy through case studies and the development of instruments for its implementation in real educational contexts.

References

Adams, E. S., Carswell, L., Kumar, A., Meyer, J., Ellis, A., Hall, P., & Motil, J. (1996). Interactive multimedia pedagogies: Report of the working group on interactive multimedia pedagogy. *ACM SIGCUE Bulletin, 28*(SI), 182–191. https://doi.org/10.1145/1013718.237646

Alruwais, N., Wills, G., & Wald, M. (2018). Advantages and challenges of using e-assessment. *International Journal of Information and Education Technology, 8*(1), 34–37. https://doi.org/10.18178/ijiet.2018.8.1.1008

Amershi, S., Carenini, G., Conati, C., Mackworth, A. K., & Poole, D. (2008). Pedagogy and usability in interactive algorithm visualizations: Designing and evaluating CIspace. *Interacting with Computers, 20*(1), 64–96. https://doi.org/10.1016/j.intcom.2007.08.003

Ardito, C., De Marsico, M., Lanzilotti, R., Levialdi, S., Roselli, T., Rossano, V., & Tersigni, M. (2004). Usability of e-learning tools. In M. F. Costabile (Ed.), *Proceedings of the Working Conference on Advanced Visual Interfaces* (pp. 80–84). Association for Computing Machinery. https://doi.org/10.1145/989863.989873

Barker, P., & King, T. (1993). Evaluating interactive multimedia courseware: A methodology. *Computers in Education, 21*(4), 307–319. https://doi.org/10.1016/0360-1315(93)90034-G
Belson, H., & Ho, J. (2012). *Usability: A fresh graduate's guide to software development tools and technologies*. School of Computing, National University of Singapore.
Bonwell, C. C., & Eison, J. A. (1991). *Active learning: Creating excitement in the classroom*. The George Washington University.
Chickering, A., & Gamson, Z. (1987). Seven principles of good practice in undergraduate education. *AAHE Bulletin, 39*, 3–7.
Elissavet, G., & Economides, A. A. (2000, December). Evaluation factors of educational software. In *Proceedings International Workshop on Advanced Learning Technologies: Advanced Learning Technology: Design and Development Issues* (pp. 113–116). IEEE.
Feizi, A., & Wong, C. Y. (2012). Usability of user interface styles for learning a graphical software application. In *2012 International Conference on Computer & Information Science (ICCIS)* (Vol. 2, pp. 1089–1094). IEEE.
Ferdig, R. E. (2006). Assessing technologies for teaching and learning: Understanding the importance of technological pedagogical content knowledge. *British Journal of Educational Technology, 37*(5), 749–760.
Gagné, R. M., Briggs, L. J., & Wager, W. W. (1988). *Principles of instructional design* (3rd ed.). Thomson Learning.
Hattie, J., & Jaeger, R. (1998). Assessment and classroom learning: A deductive approach. *Assessment in Education: Principles, Policy & Practice, 5*(1), 111–122.
Hossain, A. A. M. S. (2015). Evaluating and testing user interfaces for e-learning system: Blackboard usability testing. *Journal of Information Engineering and Applications, 5*(1), 23.
Hundhausen, C. D. (2002). Integrating algorithm visualization technology into an undergraduate algorithms course: Ethnographic studies of a social constructivist approach. *Computers and Education, 39*(3), 237–260.
Ismail, H., Hussein, N., Harous, S., & Khalil, A. (2023). Survey of personalized learning software systems: A taxonomy of environments, learning content, and user models. *Education Sciences, 13*(7), 741. https://doi.org/10.3390/educsci13070741
Issa, L., & Jusoh, S. (2019). Usability evaluation on gamified e-learning platforms. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems* (pp. 1–6).
Kagdi, H., Collard, M. L., & Maletic, J. I. (2007). A survey and taxonomy of approaches for mining software repositories in the context of software evolution. *Journal of Software Maintenance and Evolution: Research and Practice, 19*(2), 77–131.
Kehoe, C., Stasko, J., & Taylor, A. (2001). Rethinking the evaluation of algorithm animations as learning aids: An observational study. *International Journal of Human-Computer Studies, 54*(2), 265–284.
Litchfield, B. C. (1993). Design factors in multimedia environments: Research findings and implications for instructional design. *Annual Meeting of the American Educational Research Association*, 1–10.
Maulana, I. R., Firizkiansah, A., & Dewi, S. H. F. (2025). Design of a web-based book collection management information system. *Journal of Artificial Intelligence and Engineering Applications, 4*(3).
Molina, O. E., Fuentes-Cancell, D. R., & García-Hernández, A. (2022). Evaluating usability in educational technology: A systematic review from the teaching of mathematics. *LUMAT: International Journal on Math, Science and Technology Education, 10*(1), 65–88.
Nachmias, R., Mioduser, D., Oren, A., & Lahav, O. (1999). Taxonomy of educational websites: A tool for supporting research, development and implementation of web-based learning. *International Journal of Educational Telecommunications, 5*(3), 193–210.
Nahvi, M. (1996). Dynamics of student-computer interaction in a simulation environment: Reflections on curricular issues. In *Proceedings of the IEEE Frontiers in Education* (pp. 1383–1386).
Naps, T. L., Cooper, S., Koldehofe, B., Leska, C., Rößling, G., Dann, W., Korhonen, A., Malmi, L., Rantakokko, J., Ross, R. J., Anderson, J., Fleischer, R., Kuittinen, M., & McNally, M. (2003). Evaluating the educational impact of visualization. *ITiCSE*, 124–136.
Naps, T. L., Eagan, J. R., & Norton, L. L. (2000). JHAVÉ—An environment to actively engage students in web-based algorithm visualizations. In *Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education* (pp. 109–113).
Nielsen, J. (1993). *Usability engineering*. Academic Press.
Nielsen, J., & Mack, R. L. (1994). *Usability inspection methods*. John Wiley.
Norman, D. A., & Draper, S. W. (1986). *User centered system design: New perspectives on human-computer interaction*. Lawrence Erlbaum Associates.
Ormell, C. P. (1974). Bloom's taxonomy and the objectives of education. *Educational Research, 17*(1), 3–18.
Pizard, S., & Vallespir, D. (2020). Developing a taxonomy for software engineering education through an empirical approach. *CLEI Electronic Journal, 23*(2), 5–1.
Quinn, C. N. (1996). Pragmatic evaluation: Lessons from usability. In *13th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education* (pp. 15–22). Australasian Society for Computers in Learning in Tertiary Education.
Ridgway, J., McCusker, S., & Pead, D. (2004). *Literature review of e-assessment*. Futurelab.
Schweitzer, D., & Brown, W. (2007). Interactive visualization for the active learning classroom. In S. Haller, I. Russell, S. Rodger, & J. D. Dougherty (Eds.), *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (pp. 208–212). Association for Computing Machinery.
Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: Strategies for effective human-computer interaction*. Pearson Education India.
Silius, K., Tervakari, A. M., & Pohjolainen, S. (2003). A multidisciplinary tool for the evaluation of usability, pedagogical usability, accessibility and informational quality of web-based courses. In *The Eleventh International PEG Conference: Powerful ICT for Teaching and Learning* (Vol. 28, pp. 1–10).
Stamenković, S., & Jovanović, N. (2023). A web-based educational system for teaching compilers. *IEEE Transactions on Learning Technologies, 17*, 143–156.
Stamenković, S., Jovanović, N., & Chakraborty, P. (2020). Evaluation of simulation systems suitable for teaching compiler construction courses. *Computer Applications in Engineering Education, 28*(3), 606–625. https://doi.org/10.1002/cae.22231

Stamenković, S., Jovanović, N., Vasović, B., Cvjetković, M., & Jovanović, Z. (2023). Software tools for learning artificial intelligence algorithms. *Artificial Intelligence Review, 56*, 10297–10326. https://doi.org/10.1007/s10462-023-10436-0

Storey, M. A., Phillips, B., Maczewski, M., & Wang, M. (2002). Evaluating the usability of web-based learning tools. *Journal of Educational Technology & Society, 5*(3), 91–100.

Tversky, B., Morrison, J. B., & Betrancourt, M. (2002). Animation: Can it facilitate? *International Journal of Human-Computer Studies, 57*, 247–262. https://doi.org/10.1006/ijhc.2002.1017

Usman, M., Britto, R., Börstler, J., & Mendes, E. (2017). Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method. *Information and Software Technology, 85*, 43–59. https://doi.org/10.1016/j.infsof.2017.01.006

Valente, A., Otersen, S. M., & Marchetti, E. (2025). The taxonomies project: Coding as botanical art. *Letters on Informatics and Interdisciplinary Research, 6*, 1–11. https://doi.org/10.52731/liir.v006.391

Wade, V. P., & Ashman, H. (2007). Guest editors' introduction: Evolving the infrastructure for technology-enhanced distance learning. *IEEE Internet Computing, 11*(3), 16–18.