Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6, 1293-1306
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i6.8102

Bridging pedagogy and technology: A conceptual analysis for software design in educational platforms

Gabriela Pacheco Sánchez[1*], Alexandro Escudero-Nahón[2], Mauricio Arturo Ibarra Corona[3], Martín Muñoz Mandujano[4]
[1,2,3,4]Faculty of Informatics, Autonomous University of Querétaro, Mexico; gabriela.pacheco@uaq.mx (G.P.S.) alexandro.escudero@uaq.mx (A.E.N.) mauricio.ibarra@uaq.mx (M.A.I.C.) martin.munoz.mandujano@uaq.mx (M.M.M.).

Abstract: Software design is a difficult concept to define when it comes to educational technology. This is reflected in research related to the design, creation, and development of educational technology platforms. Therefore, it was decided to define the concept of software design using the conceptual cartography method through seven different axes of analysis. As a result, it was possible to identify the different conceptual components, as well as the semantic relationships of software design with other similar concepts in the field of educational technology. The need for standardization in the use and application of techniques and models for the development of educational technology platforms is mentioned. Finally, it is proposed that future work include the development of a software tool that allows the creation of platforms considering both technical and pedagogical aspects.

Keywords: *Educational software, Educational technology, Instructional design, Software design, Virtual environment.*

1. Introduction

Since the inclusion of information and communication technologies in education, multiple efforts have been made to integrate technology into the teaching-learning process in the best possible way. Thus, technology has been used in various forms, such as: a tool for course management, virtual learning environments, MOOCs, and virtual learning environments [1, 2]. Similarly, from the inclusion of technology in education, concepts such as *u-learning* have emerged, which refers to learning present in any scenario thanks to technology and the growing connectivity in the world.

However, this has posed a series of challenges that are difficult to resolve. The design and development of educational technologies is a complex task, as it requires the intervention of disciplines such as Software Engineering for the creation of these technologies, as well as other areas that contribute to the pedagogical aspects necessary for the development of the teaching-learning process [3]. This becomes particularly important and a cause for concern when considering that it is not common to find individuals who meet both profiles [4].

Thus, the increasing inclusion of educational technology in traditional teaching spaces, as well as the growth in research on educational technology, has led researchers and educators with backgrounds outside of Software Engineering to use their own models and tools for the creation of educational technology platforms [5, 6]. However, as this is a process that involves software development, it is common to find terms specific to Software Engineering used ambiguously and outside their usual semantic field [7]. Due to this, there are inconsistencies in the terminology used by authors when designing educational technology platforms [8].

In particular, the concept of "Software Design" is one that is used in various research works and, due to the aforementioned inconsistencies, it is a term that is not correctly defined within the scope of educational technology platforms. Since it is not concretely defined, the activities and responsibilities of the process of designing educational technology platforms are also unclear, making it difficult (or even

History: Received: 24 March 2025; Revised: 22 May 2025; Accepted: 26 May 2025; Published: 16 June 2025
* Correspondence: gabriela.pacheco@uaq.mx

impossible) to differentiate it from other similar terms and concepts, both in the field of educational technology and in Software Engineering.

2. Methodology

The objective of this research was to define the term "Software Design" in the context of educational technology platforms. To achieve this, the conceptual cartography method was used. Conceptual cartography allows the construction of knowledge around a concept through the review and systematization of multiple studies across eight possible axes of analysis (Notion, Categorization, Characterization, Differentiation, Classification, Linkage, Methodology, and Exemplification) [9]. Thus, the conceptual cartography was developed in four phases [10] which are described below:

2.1. Identification of the Research Problem

With the objective of this research in mind, the following research question was posed: How is the term "software design" used in research that develops educational technology platforms?

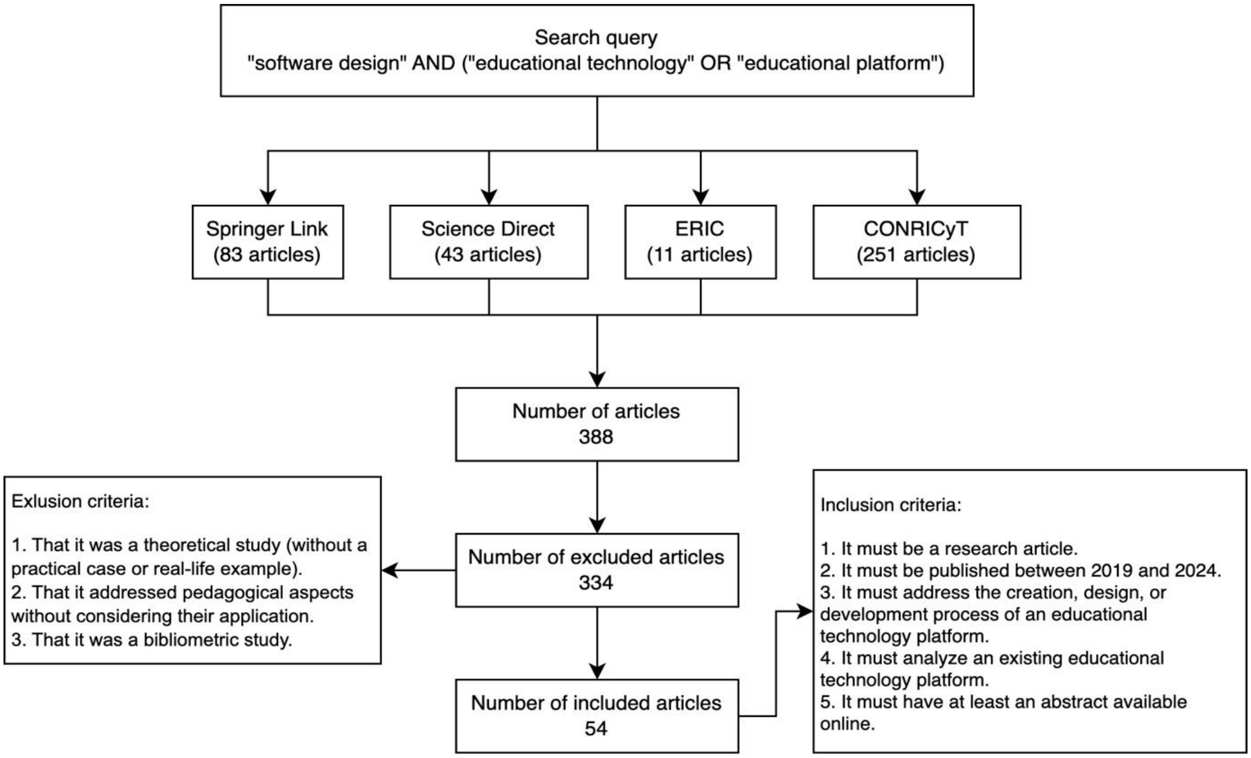
2.2. Search and Selection of Studies

For the search of documents, four scientific databases were used: CONRICyT, ERIC, ScienceDirect, and Springer Link. The same search query was used in all databases: "software design" AND ("educational technology" OR "educational platform"). Only research articles published within the five years prior to the search date and written in English or Spanish were accepted.

The following results were obtained by database: CONRICyT (251), ERIC (11), ScienceDirect (43), and Springer Link (83). With a total of 388 articles collected, which, after being compared with the inclusion and exclusion criteria (described below), were reduced to just 54.

The inclusion criteria used were: that it was a research article; that it was published between February 2019 and February 2024; that it dealt with the process of creation, design, or development of an educational technology platform; that it analyzed an existing educational technology platform; and that at least its abstract was available online. On the other hand, the exclusion criteria applied were: that it was a theoretical study (without a practical case or real example); that it discussed pedagogical aspects without considering their application; and that it was a bibliometric study. Figure 1 summarizes the search and selection process of the studies.

DOI: 10.55214/25768484.v9i6.8102


Search query
"software design" AND ("educational technology" OR "educational platform")
Springer Link (83 articles)
Science Direct (43 articles)
ERIC (11 articles)
CONRICyT (251 articles)
Number of articles 388
Number of excluded articles 334
Number of included articles 54

Exlusion criteria:

1. That it was a theoretical study (without a practical case or real-life example).
2. That it addressed pedagogical aspects without considering their application.
3. That it was a bibliometric study.

Inclusion criteria:

1. It must be a research article.
2. It must be published between 2019 and 2024.
3. It must address the creation, design, or development process of an educational technology platform.
4. It must analyze an existing educational technology platform.
5. It must have at least an abstract available online.

Figure 1.
Process of document search and selection.

2.3. Definition of the Axes of Analysis

For the purposes of the research and based on the research question, it was determined that only seven of the eight axes of analysis were relevant to achieve the initially stated objective. The axes used, along with their respective central questions, are presented in Table 1.

Table 1.
Axes of the conceptual cartography of the concept "Software Design". Source: Own elaboration based on.

Axis of Analysis	Central Question
1. Notion	What does "designing software" mean in the context of educational technology platforms?
2. Categorization	To which field does the concept of software design belong?
3. Characterization	What are the main characteristics of the concept of software design when it comes to educational technology platforms?
4. Differentiation	What other similar concepts exist, and how does it differ from them?
5. Linkage	How is software design related to other techniques, tools, and methodologies for the creation of educational technology platforms?
6. Methodology	What elements make up a methodology for software design of educational technology platforms?
7. Exemplification	What relevant examples exist regarding the application of the concept of software design in educational technology platforms?

Source: Escudero-Nahón [9] and Lee and Kim [10]

2.4. Cartographic Analysis

With the axes defined, it was possible to classify the documents obtained in the previous stage according to their relevance and pertinence in answering the central question. It was allowed for the same article to be used to answer multiple axes of analysis. Table 2 presents the number of articles used to answer each central question.

Table 2.
Articles used per axis of analysis. Source: Own elaboration.

Axis of Analysis	Number of Articles
1. Notion	23
2. Categorization	8
3. Characterization	31
4. Differentiation	5
5. Linkage	20
6. Methodology	8
7. Exemplification	12

Once the classification was done, it was possible to carry out the conceptual analysis of the term "Software Design" across its different axes. This analysis is presented in the following section.

3. Results

3.1. Notion: What Does "Designing Software" Mean in the Context of Educational Technology Platforms?

According to multiple authors, the concept of "designing software" is a collaborative and multidisciplinary process involving teachers, students, engineers, and designers (or at least considering their perspectives) [11-13].

Recent work by Ifenthaler and Pellegrino [14] formalizes this collaboration through their Co-Design Framework for Educational Technology, which emphasizes iterative prototyping with joint teacher-developer teams to align pedagogical goals with technical implementation. However, at this point, it is possible to find different positions on the matter, as there is no consensus on the specific tasks involved in designing an educational technology platform. The term 'design' in educational technology spans a broad spectrum, from accessible pedagogical decisions to complex technical implementations [15]. Designing learning technologies involves layered responsibilities, where some team members focus on usability while others handle backend engineering challenges [16].

This represents a problem, as it is complex to orchestrate all the activities involved in software design, especially if they are treated as separate tasks without a defined process. This is mentioned and debated from three perspectives, which propose different approaches to what designing an educational technology platform entails. The first perspective argues that educational technology design should prioritize pedagogical considerations, focusing on instructional design within digital platforms rather than their technical development [17-19].

This perspective removes everything related to software design and creation from the equation, simplifying the semantic complexity by limiting the design process to the configuration and application of an existing platform. Under this perspective, by not considering the development of the platform, problems arise from the low cohesion between the objectives of the teachers using the platform and what it offers.

The second perspective is diametrically opposed to the first, as it proposes that software design (when it comes to educational technology platforms) should be viewed and applied from a technical standpoint, oriented exclusively towards the processes, methods, and techniques offered by software engineering [20-24]. This initial standpoint advocates for prioritizing teaching and learning principles over technological features when developing digital education tools. In essence, the focus shifts to how instructional methods are operationalized within digital environments, rather than the technical engineering of the platforms themselves. In this context, pedagogical and instructional design aspects are addressed through requirements engineering, but teachers or students are not involved in the design process. Here, the goal is to establish and meet supplementary characteristics of the platform, such as scalability, portability, availability, and security. While this ensures technical quality, it risks misaligning with end-user needs [25-27].

Despite the opposing approaches of the aforementioned perspectives and their vision of what software design is, they share some similarities. The main one, perhaps, lies in the fact that by focusing solely on one aspect (technical or instructional), the other is neglected, leading to similar problems, such as the lack of alignment between the objectives a platform can help achieve and the needs of the users. Thus, the third perspective seeks to combine the best aspects of the first two.

Therefore, the third perspective proposes that software design (when it comes to educational technology platforms) should be treated as a process that considers technical, technological, and pedagogical aspects, so that there are no gaps between technology and pedagogy [28-32]. It makes sense to think that this task needs to be carried out in a multidisciplinary manner, as the technical deficiencies that teachers may have can be covered by the experience that software designers and developers possess.

In this way, this last perspective establishes that in software design, it is not enough to simply consult experts in pedagogy and instructional design, but it is necessary to integrate them actively into the design process, allowing the formation of teachers who are also designers [6]. Currently, answering the question 'What does designing software mean in educational technology contexts?' remains complex, as it involves a multi-stage process requiring active participation from all key stakeholders (teachers, students, and developers) to ensure alignment between pedagogical objectives, user needs, and technical system requirements [33].

Thus, it is possible to say that, in the context of educational technology platforms, designing software means developing the learning objects to be used in the platform, carrying out the instructional design to be implemented, and, of course, considering the technical aspects of the platform, such as its architecture and the information structures to be used.

3.2. Categorization: To Which Field Does the Concept of Software Design Belong?

The concept of software design originated in the context of computer science, specifically in software engineering. As part of the software development process, design is a fundamental stage in the planning and definition of a system [34, 35]. Recent work by Zhang and Zhang [1] demonstrates how this foundational concept has evolved, showing that modern educational software design now requires "pedagogical validity patterns" (p. 15) to bridge technical and instructional requirements. Since its conception, due to the constant and growing inclusion of technology in multiple areas of knowledge, this concept has been adapted and used according to the purposes it seeks to fulfill.

Education is one of the areas of knowledge that most recently integrated technology, resulting in what is now known as educational technology, within which multiple branches can be found, such as educational technology platforms, gamification, u-learning, e-learning, and, of course, the design of educational technology [36-39]. Thus, it is possible to mention that it is a polymorphic concept, adapted according to the convenience of those who use it, which generates confusion regarding what this term originally implies (at least in the context of educational technology).

Therefore, this term is sometimes mistakenly interchanged and compared with concepts such as instructional design, learning object design, and virtual course design. Terms that, although they share some similarities, are very different in their origin, function, and scope. This only generates ambiguity regarding the scope and limits of the concept.

However, from the perspective of educational technology, software design can refer to the design of pedagogical aspects or the design of technical aspects [2]. The former suggests the planning of the properties of a platform that will allow the proper development of the teaching-learning process [2, 28, 39, 40]. On the other hand, the latter refers to the planning of the quality attributes of the software and how they will be met, addressing aspects such as usability, availability, scalability, and portability [20, 23, 24].

In this context, it becomes evident that conceptual definitions must be carefully adapted to the semantic framework of their respective knowledge domains. Yet a persistent contradiction emerges when terminology is adopted nominally while disregarding its foundational theoretical underpinnings. This conceptual dissonance has prompted scholars to question whether educational technology warrants the establishment of a specialized software engineering subdiscipline [41, 42].

3.3. Characterization: What are the Main Characteristics of the Concept of Software Design When it Comes To Educational Technology Platforms?

When planning and developing an educational technology platform, it is possible to observe software design through multiple conceptual lenses. These development processes inevitably reveal the complex interplay between pedagogical design requirements and technological implementation considerations [43]. It is interesting how software design can encompass all the aspects that an educational technology project involves, but the lack of cohesion between these aspects is also clear [29].

In some ways, although both aspects are considered, each has its own characteristics that must be taken into account for successful design [44, 45]. Table 3 presents the elements that characterize software design, as well as their understanding and representation according to the aspect being analyzed.

Table 3.
Comparison between the design of aspects considered by software design in educational technology platforms. Source: Own elaboration based on.

Element	Pedagogical Aspect	Technological Aspect
Object of Study	Learning objects, instructional design models, learner engagement strategies	Software architecture, development frameworks, UX/UI design
Objective	Align platform features with learning theories and pedagogical goals	Ensure scalability, security, and technical robustness
Evaluation	Learning analytics, student performance metrics	A/B testing, system performance benchmarks
Quality Metrics	Learning effectiveness, accessibility	Load capacity, response time, cross-device compatibility
Design Language	Pedagogical design patterns	UML, SysML
Key Actors	Educators, instructional designers	Software architects, DevOps engineers
Process Model	ADDIE, backward design	Agile/Scrum, CI/CD pipelines

Source: Zawacki-Richter, et al. [46]

After analyzing the characteristics of software design in the context of educational technology platforms, it is possible to identify the strong separation between the pedagogical and technological aspects. The difference in objectives is clear evidence that, although software design can encompass everything involved in the development of a platform, there is no unification or common model that allows aligning the activities that must be carried out to create a useful, usable, and functional product that also allows the proper development of the teaching-learning process [47].

3.4. Differentiation: What Other Similar Concepts Exist, and How Does it Differ from Them?

When developing an educational technology platform, it is possible to identify two concepts similar to software design, which are used to a greater or lesser extent by authors depending on the objective they seek to achieve. These are: instructional design and digital instructional design. The first is used when the platform is oriented towards the creation of pedagogical resources and instructional content, while the second is used when the goal is to transfer a learning environment to a virtual setting.

These concepts, although similar, have substantial differences when analyzing the purpose they seek to fulfill. Khan and Bell [48] "systematic review of 215 EdTech projects revealed that projects failing to properly distinguish between these concepts had 32% lower adoption rates" (p. 7). To clarify the divergences between the three concepts, Table 4 presents a comparison across seven aspects shared by these notions.

Table 4.
Comparison between concepts similar to software design. Source: Own elaboration based on.

Aspect	Software Design	Instructional Design	Digital Instructional Design
Abstraction	Encapsulation of concepts that allows simplifying a concept while reducing its theoretical complexity and increasing its ease of understanding.	Process of modeling instructional activities at a high level to reduce complexity and increase efficiency.	Focus on the main ideas of a learning topic, retaining abstract information in semantic forms through visual models of the course.
Modularity	Division of a software project into parts, allowing its development in stages, reducing complexity and facilitating its development.	Process of segmenting instructional content into small modules to manage complexity and maintainability.	Division of content into learning units of a course.
Reusability	Design principle that seeks the creation of generic, modifiable, and adaptable software components according to the context.	Ability to reuse existing instructional content in different applications with minimal changes.	Maximization of the cost-effectiveness of learning and performance through the reuse and dissemination of previously designed instructional materials.
Compatibility	Design property of a software component that indicates the degree of difficulty in interacting with other components.	Ability of instructional content to be used with other resources without modification.	Accessibility of learning materials from multiple devices.
Extensibility	Ability of a software component to be extended with new functionalities without major changes to the design.	Ability to add new content and materials without making major changes to the design.	Ability of a learning unit to be used for the learning and contextualization of other thematic content.
Scalability	Property of a software product that indicates the ease with which it can be used on a larger scale than its current one.	Ability of the design to be used in environments with a growing number of people.	Scaling of traditional learning means through the massification of learning units.
Maintainability	Ability of a software product to be maintained easily and periodically without the need for corrections.	Measure that indicates the ease with which materials and content can be added to the design within a stipulated time frame.	Design of clear, concise, and concrete learning units that allow their modification without affecting the course as a whole.

Source: Vallis, et al. [7]; Ifenthaler [49]; Krouska, et al. [50]; Papadakis [51]; Maulidiya, et al. [52]; Sáiz Manzanares, et al. [53]; Tlili, et al. [54]; and González-Hernández, et al. [55].

From this comparison, it is clear that the main difference between the concepts lies in the object of study. While software design aims to ensure the quality of the product to be developed, instructional design seeks the quality of the elements, components, and learning materials, and learning design seeks, as its name suggests, the quality of the learning process through the clear establishment of goals.

This seems to indicate that software design is focused solely on technical aspects. However, it is evident that, to fulfill its purpose, it encompasses all the necessary aspects for an educational technology platform. Thus, software design necessarily encompasses the other two concepts.

3.5. Linkage: How is Software Design Related to Other Techniques, Tools, and Methodologies For the Creation of Educational Technology Platforms?

Software design is related to multiple concepts within and outside the context of educational technology. Among these are: instructional design, *u-learning*, and agile development methodologies [56, 57]. First two concepts belong to the field of educational technology, while the latter is part of software engineering. Thus, designing software is a concept that, due to its relevance and ability to be used in practically any field, is adapted and used according to the needs it seeks to satisfy [58, 59].

Instructional design is a discipline oriented towards the design, definition, and planning of the activities and resources that will be developed in a course [60, 61]. Usually, in this task, elements such as the content to be reviewed, the teaching methodology, communication channels, and, more recently, the digital tools to be used (management tools, audiovisual media, platforms, etc.) are defined. This term is directly related to software design by defining, in the course design, the characteristics that the digital tool or platform requires [7, 20].

Similarly, *u-learning* proposes the presence of learning opportunities in any environment and at any time through information technologies. It is usually presented as a personalized learning style tailored to the needs and capabilities of the student [62, 63]. It is supported by gamification (usually through mobile devices), while proposing characteristics that educational platforms must consider [64-67].

On the other hand, agile development methodologies serve as a guide for the proper development of software products. The 2024 State of EdTech Development Report found that teams using pedagogically-adapted Agile (EduScrum 2.0) reduced development time by 30% while improving instructor satisfaction by 45% [68]. Typically, these methodologies require advanced technical expertise and are often driven by stakeholder requirements rather than end-user needs, which can lead to poor platform adoption [69-71]. Recent studies highlight that even in agile environments, neglecting user-centered design principles results in lower acceptance rates [72, 73]. However, as a reference framework, it serves software design by providing a structure into which other techniques, tools, and methodologies (such as instructional design and *u-learning*) can be integrated with the aim of increasing the chances of success of the final product.

In this way, software design is currently related to concepts specific to education and software engineering. However, it has the potential to be linked and extended with other tools, techniques, and methodologies, as it has been established previously, it is a flexible concept that can be standardized as needed.

3.6. Methodology: What Elements Make Up a Methodology for Software Design of Educational Technology Platforms?

Talking about a methodology implies the existence of a structured and concrete process. However, when it comes to the development of educational technology, it becomes complex due to two factors: the flexibility required in a teaching-learning process and the number of variants and variables that influence the acquisition of new knowledge. Recent studies highlight that student and teacher engagement in EdTech platforms remains a persistent challenge, often due to insufficient integration of pedagogical principles into design frameworks [74].

Despite this, there are also success cases, where one of the constants has been the structured application of a development methodology. Thanks to this, it is possible to identify and analyze the characteristics that determine the success or failure of a project of this type. Undoubtedly, a predominant characteristic in cases with positive results is that it is an iterative and continuous improvement process. This implies a cyclical and incremental structure, in which the stated objectives are met in stages [57].

Similarly, the most successful implementations consistently incorporate two key design elements: active stakeholder participation (primarily teachers and students), and substantial time investment in usability and user experience design [75-77]. Figure 2 presents a diagram that groups the elements and characteristics that have been considered in different instances with positive results.

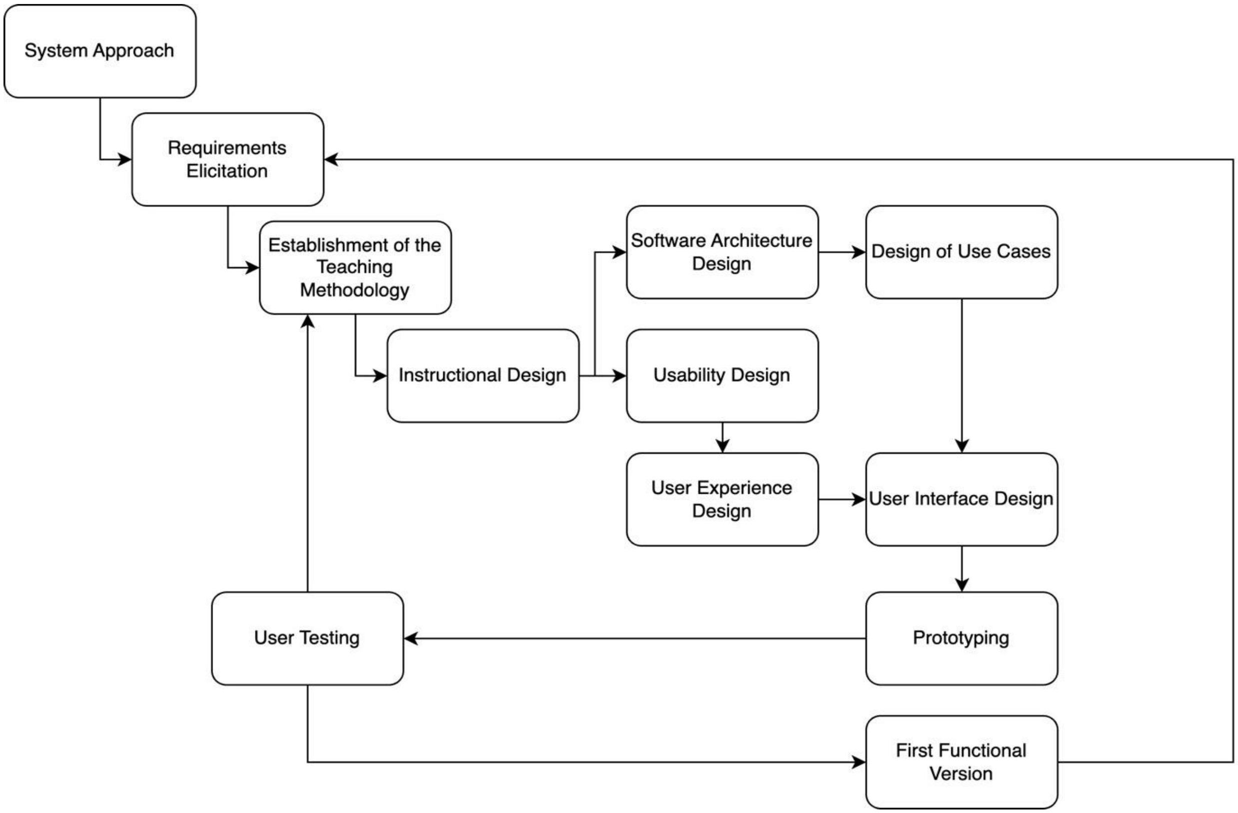


Figure 2.
Design process of educational technology platforms. Source: Own elaboration based on.
Source: Ifenthaler [78]

Contemporary educational technology platforms require rigorous integration of evidence-based pedagogical elements. Modern instructional design frameworks now systematically incorporate: learning sciences principles, adaptive active learning methodologies, and embedded formative assessment [28, 79, 80]. These core components establish the methodological foundation guiding both platform development and evaluation protocols.

Exemplification: What relevant examples exist regarding the application of the concept of software design in educational technology platforms?

As described, examples of software design applied in the process of creating an educational technology platform vary in their approach. The first approach is technical, where tools specific to software engineering are used, and tools and models related to education are integrated. Zhang and Chen [81] analysis of GitHub repositories revealed that 68% of technically-driven EdTech projects now employ AI-assisted coding tools, but only 29% incorporate pedagogical validation in their development pipelines. While knowledge from both disciplines is integrated, by focusing on aspects such as functionality, software architecture, and usability, the pedagogical aspect is often left in the background. The works of Hernández, et al. [82] and Alam and Shi [83] and newer examples like Li, et al. [84] automated learning analytics platform demonstrate this technical focus, achieving 99.8% system reliability but limited educator adoption.

On the other hand, it is possible to identify a second variant of application, where the technical aspect is relegated to give greater relevance to the pedagogical content and instructional design of the platform. In this type of case, the supplementary characteristics of a platform (scalability, usability, functionality, extensibility, etc.) are not considered in the development of the platform. This results in a platform that is successful in very specific cases, as it is practically impossible to transfer or adapt it to other domains or contexts. In this type of application, the works of Radović, et al. [85] and O'Dell [86] are present.

Finally, there is a third variant in which software design is adapted to be used alongside instructional design techniques and teaching methodologies. This approach emerges as a solution to reconcile the semantic and conceptual differences between technical disciplines and education. Contemporary examples of this integrated approach can be found in the works of Chimalakonda and Nori [43]; Turk, et al. [40]; Ifenthaler and Tracey [30]; Karoui, et al. [21]; Martínez, et al. [32].

The contrast between the approaches is due to multiple factors, such as the objective and context of the research. However, it is undeniable that the academic and professional background of the authors directly influences the approach when designing an educational technology platform. On one hand, those with a technical background opted to use tools and techniques specific to the field of information technologies, while those with a background in teaching and pedagogy used models and methodologies present in the educational field. However, it is interesting that the authors who integrated both disciplines equally do not have specific training, although they do mention interdisciplinary work in their studies.

4. Conclusions

In conclusion, defining software design remains complex due to conceptual breadth and ambiguity about its constituent activities. While various perspectives exist for educational technology development, contemporary research confirms its inherently multidisciplinary nature [43]. Paradoxically, despite this recognized need for interdisciplinary collaboration, integrated team approaches remain exceptional in practice [32]. While this would indicate that it is necessary to involve experts from various fields to cover the technological and pedagogical aspects that the design of educational technology platforms entails, the reality is that this is uncommon.

Usually, it is a single person (or group of people) who is responsible for the design process and everything it entails (platform planning, technology selection, interface design, etc.). And while it is not uncommon to find this type of situation in software development projects, the reality is that multidisciplinary training is necessary, which many lack [5].

Instructional design and software design share several elements in common. Both are (mostly) iterative processes, consider the target audience, and have reference models and support tools. However, semantic differences prevent authors from integrating the concepts and techniques of one field of knowledge with another, despite the similarities between the two concepts. Due to this, the need to generate models that align both disciplines has been raised, or even to create a specific branch of Software Engineering to address the design of educational platforms [20].

Now, This need faces significant implementation challenges that are easier to theorize than actualize. These obstacles range from the inherent complexity and diversity of teaching-learning processes to the scalability requirements across different instructional contexts [43]. For now, although there are already some ideas about models that consider both disciplinary areas and that help support pedagogy in educational platforms, there is nothing regarding the supplementary characteristics of the software, which are essential for ensuring the quality of the final product [85, 86].

Thus, in response to the question, "How is the term 'software design' used in research that develops educational technology platforms?" Currently, it is used ambiguously, with a meaning and implications dependent on the professional background of the person using it. However, it is noteworthy that as technology becomes increasingly embedded in teaching-learning processes, a growing body of research recognizes the essential multidisciplinary nature of educational technology design [32]. This, together with existing models, allows the formation of teachers as designers [6].

Therefore, software design, as a concept in the development of educational technology platforms, is composed of concepts from different semantic branches. On one hand, there are those that refer to instruction, the design of a course according to its different stages, learning objectives, and target audience, while on the technological side, there is talk of design stages, platform user objectives, and, finally, the end users of the platform.

Due to the emergence and popularity of concepts such as u-learning, e-learning, gamification, and digital instructional design, it is feasible to think that, over time, teachers will acquire the multidisciplinary training required for the design of platforms and educational software. Evidence of this is the existence and growing use of universal modeling languages for digital instructional design (coUML, derived from the unified language specific to Software Engineering).

From the above, it is possible to say that it would be necessary not only to think of a reference model that unifies both disciplines but also to think of a software tool that facilitates the generation of the elements required by educational technology platforms.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] Y. Zhang and X. Zhang, "The impact of online interaction and information technology accessibility on academic engagement among international undergraduate students in Chinese universities: The mediating effect of learning interest," *Acta Psychologica*, vol. 249, p. 104478, 2024. https://doi.org/10.1016/j.actpsy.2024.104478

[2] M. Deev and A. Finogeev, "Application of the convergent education model in the development of a smart learning environment," *Telematics and Informatics Reports*, vol. 10, p. 100051, 2023. https://doi.org/10.1016/j.teler.2023.100051

[3] M. M. Saldívar-Almorejo, L. A. Flores-Herrera, R. Rivera-Blas, P. A. Niño-Suárez, E. Z. Rivera-Blas, and N. Rodríguez-Contreras, "e-Learning Challenges in STEM Education," *Education Sciences*, vol. 14, no. 12, p. 1370, 2024. https://doi.org/10.3390/educsci14121370

[4] K. C. Li and B. T.-M. Wong, "Review of smart learning: Patterns and trends in research and practice," *Australasian Journal of Educational Technology*, vol. 37, no. 2, pp. 189-204, 2021. https://doi.org/10.14742/AJET.6617

[5] J. Oudenampsen, E. Das, N. Blijlevens, and M. van de Pol, "How to cross the line: Design principles for interdisciplinary education," *MedEdPublish*, vol. 13, p. 35, 2024. https://doi.org/10.12688/mep.19693.2

[6] K. H. Rubilar, "Design for learning: Teacher educators as policymakers?," *School-University Partnerships*, 2025. https://doi.org/10.1108/SUP-10-2024-0023

[7] C. J. Vallis, H. T. Nguyen, and A. Norman, "Cross-cultural adaptation of educational design patterns at scale," *Journal of Work-Applied Management*, vol. 16, no. 2, pp. 253-268, 2024. https://doi.org/10.1108/JWAM-10-2023-0106

[8] J. Muñoz-Arteaga, G. C. López-Torres, and Á. E. Muñoz-Zavala, "An agile learning methodology to support inclusive education: An agile learning methodology to support inclusive education," *Edutec, Revista Electrónica de Tecnología Educativa*, vol. 86, pp. 116-136, 2023. https://doi.org/10.21556/edutec.2023.86.3013

[9] A. Escudero-Nahón, "Intermodalidad educativa: propuesta de desarrollo conceptual con una revisión sistemática y una cartografía conceptual," *Desafíos educativos*, vol. 3, no. 6, pp. 19-28, 2019.

[10] J. Lee and S. Kim, "A five-phase model for dynamic conceptual mapping in educational technology," *Educ. Res. Rev*, vol. 38, p. 100532, 2023. https://doi.org/10.1016/j.edurev.2023.100532

[11] T. Almeida, T. Pessoa, and J. Bidarra, "Co-designing educational software: A case study of teacher-developer collaboration," *J. Educ. Technol. Soc*, vol. 26, no. 2, pp. 45–58, 2023. https://doi.org/10.30191/ETS.202304_26(2).0004

[12] C. Kulkarni *et al.*, "Peer and self assessment in massive online classes," *ACM Transactions on Computer-Human Interaction*, vol. 20, no. 6, pp. 1-31, 2013. https://doi.org/10.1145/2505057

[13] L. P. Prieto, K. Sharma, Ł. Kidzinski, and P. Dillenbourg, "Orchestration load indicators and patterns: In-the-wild studies using mobile eye-tracking," *IEEE Transactions on Learning Technologies*, vol. 11, no. 2, pp. 216-229, 2017.

[14] D. Ifenthaler and J. W. Pellegrino, "Bridging the pedagogy-technology divide: A framework for co-designing educational software," *Comput. \& Educ*, vol. 210, p. 104960, 2024.

[15] D. Hernández-Leo, R. Martinez-Maldonado, A. Pardo, J. A. Muñoz-Cristóbal, and M. J. Rodríguez-Triana, "Designing for the boundaries of learning technology ecosystems," *Comput. Educ*, vol. 168, p. 104183, 2021. https://doi.org/10.1016/j.compedu.2021.104183

[16] A. F. Wise, S. Knight, and S. B. Shum, "The expanding role of design in learning engineering," *J. Learn. Anal*, vol. 10, no. 1, pp. 45–62, 2023. https://doi.org/10.18608/jla.2023.7892

[17] D. Ifenthaler, D. Gibson, and D. Prasse, "Theoretical foundations of learning design for 21st century competencies," *Technol. Knowl. Learn*, vol. 28, no. 1, pp. 5–27, 2023. https://doi.org/10.1007/s10758-022-09612-y

[18] V. Kovanović, S. Joksimović, and D. Gašević, "Learning design: Conceptual and computational support for contemporary learning scenarios," *Comput. Human Behav*, vol. 137, p. 107398, 2022.

[19] M. MacLeod, H. H. Yang, and Y. Shi, "Pedagogical design perspectives in digital learning ecosystems: Who owns the pedagogy?," *Educ. Technol. Res. Dev*, vol. 69, no. 2, pp. 1003–1026, 2021.

[20] S. Chimalakonda and K. V. Nori, "An ontology based modeling framework for design of educational technologies," *Smart learning environments*, vol. 7, no. 1, p. 28, 2020. https://doi.org/10.1186/s40561-020-00135-6

[21] A. Karoui, M. Guerrero, and M. Tremblay, "Toward an educational software engineering discipline," *IEEE Trans. Educ*, vol. 63, no. 4, pp. 318–327, 2020.

[22] L. D. A. Almeida, R. M. de Souza, and K. M. de Oliveira, "A systematic process for domain-specific software design in education," *J. Syst. Softw*, vol. 173, p. 110872, 2021. https://doi.org/10.1016/j.jss.2020.110872

[23] M. C. Gómez-Álvarez, J. M. Cueva-Lovelle, and O. Sanjuán-Martínez, "Engineering principles for designing adaptive learning platforms," *IEEE Access*, vol. 10, pp. 45672–45685, 2022. https://doi.org/10.1109/ACCESS.2022.3161130

[24] P. Singh, S. Sharma, and P. Kaur, "A model-driven approach for designing educational software with formal validation," *Educ. Inf. Technol*, vol. 28, no. 1, pp. 789–812, 2023.

[25] S. Almalki and R. Aziz, "The gap between technical and pedagogical requirements in EdTech platforms: A developer perspective," *Educ. Technol. Res. Dev*, vol. 71, no. 2, pp. 345–362, 2023. https://doi.org/10.1007/s11423-023-10203-6

[26] I. M. Gómez-Trigueros, M. Ruiz-Bañuls, and D. Ortega-Sánchez, "Technical versus pedagogical quality in digital learning environments: An unresolved tension," *Comput. Educ*, vol. 188, p. 104563, 2022. https://doi.org/10.1016/j.compedu.2022.104563

[27] H. Sadeghi, R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *Journal of Systems and Software*, vol. 172, p. 110851, 2021. https://doi.org/10.1016/j.jss.2020.110851

[28] C. Xieling, D. Zou, X. Haoran, and W. F. Lee, "Past, present, and future of smart learning: a topic-based bibliometric analysis," *International Journal of Educational Technology in Higher Education*, vol. 18, no. 1, pp. 1-29, 2021. https://doi.org/10.1186/s41239-020-00239-6

[29] M. G. Jamil and S. O. Isiaq, "Teaching technology with technology: Approaches to bridging learning and teaching gaps in simulation-based programming education," *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, p. 25, 2019. https://doi.org/10.1186/s41239-019-0159-9

[30] D. Ifenthaler and M. W. Tracey, "Learning design and technology: Bridging the gap between theory and practice," *Educ. Technol. Res. Dev*, vol. 70, no. 3, pp. 789–792, 2022.

[31] A. Klašnja-Milićević, M. Ivanović, and Z. Budimac, "Data science in education: Big data and learning analytics," *Computer Applications in Engineering Education*, vol. 25, no. 6, pp. 1066-1078, 2017.

[32] R. Martínez, A. Fernández, and M. Gómez, "Holistic design of EdTech platforms: Balancing pedagogy, usability, and scalability," *Int. J. Educ. Technol. High. Educ*, vol. 20, no. 1, p. 15, 2023.

[33] R. Luckin, M. Cukurova, C. Kent, and B. Du Boulay, "Empowering educators to be AI-ready," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100076, 2022. https://doi.org/10.1016/j.caeai.2022.100076

[34] I. Sommerville, *Software engineering*. Chicago: Pearson, 2020.

[35] R. S. Pressman and B. R. Maxim, *Software engineering: A practitioner's approach*. McGraw-Hill, 2020.

[36] O. Zawacki-Richter and C. Latchem, "Exploring four decades of research in Computers & Education," *Computers & Education*, vol. 122, pp. 136-152, 2018. https://doi.org/10.1016/J.COMPEDU.2018.04.001

[37] A. Bozkurt *et al.*, "A global outlook on the futures of education," *Distance Educ*, vol. 44, no. 1, pp. 1–20, 2023.

[38] C. Alario-Hoyos, I. Estévez-Ayres, M. Pérez-Sanagustín, C. D. Kloos, and C. Fernández-Panadero, "Trends in adaptive learning systems," *IEEE Trans. Learn. Technol*, vol. 15, no. 2, pp. 198–213, 2022. https://doi.org/10.1109/TLT.2022.3153009

[39] M. Zamiri and A. Esmaeili, "Methods and technologies for supporting knowledge sharing within learning communities: A systematic literature review," *Administrative Sciences*, vol. 14, no. 1, p. 17, 2024. https://doi.org/10.3390/admsci14010017

[40] M. Turk, B. C. Heddy, and R. W. Danielson, "Teaching and social presences supporting basic needs satisfaction in online learning environments: How can presences and basic needs happily meet online?," *Computers & Education*, vol. 180, p. 104432, 2022.

[41] K. Nori, W. Adnan, and M. Asadi, "Rethinking software engineering for education: A domain-specific approach," *ACM Trans. Comput. Educ*, vol. 21, no. 3, pp. 1–25, 2021.

[42] S. Chimalakonda, "Educational software engineering: A manifesto for change," *J. Educ. Technol. Syst*, vol. 51, no. 3, pp. 345–367, 2023.

[43] S. Chimalakonda and K. V. Nori, "A patterns approach for modeling educational technologies," *IEEE Trans. Learn. Technol*, vol. 16, no. 2, 2023. https://doi.org/10.1109/TLT.2022.3161130

[44] S. Amiri, M. K. Saberi, and O. K. Hussain, "Balancing pedagogical and technical requirements in educational software design," *J. Educ. Comput. Res*, vol. 58, no. 5, pp. 987–1012, 2020. https://doi.org/10.1177/0735633120907221

[45] R. E. García, C. M. Fernández, and P. L. Martínez, "Dual perspectives in EdTech development: Pedagogical needs vs. technical constraints," *Comput. Educ*, vol. 185, p. 104521, 2022.

[46] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education–where are the educators?," *International journal of educational technology in higher education*, vol. 16, no. 1, pp. 1-27, 2019.

[47] N. A. Koehler, A. D. Thompson, A.-P. Correia, and L. S. Hagedorn, "Designing online software for teaching the concept of variable that facilitates mental interaction with the material: systemic approach," *Educational Technology Research and Development*, vol. 63, pp. 97-124, 2015. https://doi.org/10.1007/s11423-014-9357-2

[48] R. Khan and T. Bell, "Conceptual clarity in EdTech design: A systematic review," *Educ. Technol. Res. Dev*, vol. 72, no. 1, pp. 1–24, 2024.

[49] D. Ifenthaler, "Learning analytics design," in learning, design, and technology," Springer, 2019, pp. 1–23.

[50] A. Krouska, C. Troussas, and M. Virvou, "Comparative evaluation of algorithms for sentiment analysis over social networking services," *J. Univers. Comput. Sci.*, vol. 23, no. 8, pp. 755-768, 2017.

[51] S. Papadakis, "Advances in mobile learning educational research (AMLER): Mobile learning as an educational reform," *Advances in Mobile learning educational research*, vol. 1, no. 1, pp. 1-4, 2021.

[52] D. Maulidiya, B. Nugroho, H. B. Santoso, and Z. A. Hasibuan, "Thematic evolution of smart learning environments, insights and directions from a 20-year research milestones: A bibliometric analysis," *Heliyon*, vol. 10, p. e26191, 2024. https://doi.org/10.1016/j.heliyon.2024.e26191

[53] M. C. Sáiz Manzanares, R. Marticorena Sánchez, C. I. García Osorio, and J. F. Díez-Pastor, "How do B-learning and learning patterns influence learning outcomes?," *Frontiers in Psychology*, vol. 8, p. 745, 2017.

[54] A. Tlili, D. Burgos, R. Huang, S. Mishra, R. C. Sharma, and A. Bozkurt, "The evolution of educational technology during COVID-19: A bibliometric analysis," *Interact. Learn. Environ*, 2022.

[55] F. González-Hernández, R. Zatarain-Cabada, M. L. Barrón-Estrada, and H. Rodríguez-Rangel, "Recognition of learning-centered emotions using a convolutional neural network," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 5, pp. 3325-3336, 2018.

[56] B. Gros and F. J. García-Peñalvo, "Future trends in the design strategies and technological affordances of e-learning," in learning, design, and technology, J. M. Spector, B. B. Lockee, and M. D. Childress, Eds." Cham: Springer, 2023.

[57] Q. N. Naveed, M. M. Alam, A. I. Qahmash, and K. M. Quadri, "Exploring the determinants of service quality of cloud e-learning system for active system usage," *Applied Sciences*, vol. 11, no. 9, p. 4176, 2021. https://doi.org/10.3390/app11094176

[58] S. Papadakis, "Cross-domain software engineering in educational contexts," *Comput. \& Educ*, vol. 185, p. 104347, 2022.

[59] M. J. Rodríguez-Triana, A. Martínez-Monés, and S. Villagrá-Sobrino, "Bridging educational needs and technological solutions: A systematic review of learning design tools," *IEEE Trans. Emerg. Top. Comput*, vol. 8, no. 2, pp. 456–470, 2020.

[60] R. M. Branch and İ. Varank, *Instructional design: The ADDIE approach*. New York: Springer, 2009.

[61] J. Stefaniak, "The instructional design trainer's guide: Authentic practices and considerations for mentoring ID and ed tech professionals," 2021.

[62] H. Xie, H.-C. Chu, G.-J. Hwang, and C.-C. Wang, "Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017," *Computers & Education*, vol. 140, p. 103599, 2019. https://doi.org/10.1016/j.compedu.2019.103599

[63] L. H. Wong *et al.*, "IDC theory: Creation and the creation loop," *Research and Practice in Technology Enhanced Learning*, vol. 14, pp. 1-29, 2019.

[64] S. Deterding, "Gamification in management: Between choice architecture and humanistic design," *J. Manag. Inq*, vol. 28, no. 2, pp. 131–136, 2019. https://doi.org/10.1177/1056492618790913

[65] J. Koivisto and J. Hamari, "The rise of motivational information systems: A review of gamification research," *International journal of information management*, vol. 45, pp. 191-210, 2019.

[66] R. Huang, A. Tlili, T. W. Chang, X. Zhang, F. Nascimbeni, and D. Burgos, "Disruptive technologies and pedagogical innovations in education," *Sustainability*, vol. 12, p. 1531, 2020.

[67] Z. Zainuddin, S. K. W. Chu, M. Shujahat, and C. J. Perera, "The impact of gamification on learning and instruction: A systematic review of empirical evidence," *Educational research review*, vol. 30, p. 100326, 2020.

[68] M. Garcia-Ruiz, P. Santos, J. A. Smith, L. Chen, and R. V. O'Connor, "EduScrum 2.0: The 2024 state of EdTech development," *J. Syst. Softw*, vol. 210, p. 111893, 2024. https://doi.org/10.1016/j.jss.2024.111893

[69] D. E. Strode, S. L. Huff, and A. Tretiakov, "The impact of organizational culture on agile method use," presented at the In 2009 42nd Hawaii International Conference on System Sciences (pp. 1-9). IEEE. https://doi.org/10.1109/HICSS.2009.436, 2009.

[70] R. Hoda, N. Salleh, and J. Grundy, "The rise and evolution of agile software development," *IEEE Software*, vol. 35, no. 5, pp. 58-63, 2018. https://doi.org/10.1109/MS.2018.290111318

[71] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements challenges in agile software development," *Requir. Eng*, vol. 27, no. 1, pp. 89–114, 2022. https://doi.org/10.1016/j.infsof.2019.01.009

[72] F. Fagerholm, A. Hellas, and J. Munch, "Developer experience in agile teams: A survey," *J. Syst. Softw*, vol. 172, p. 110852, 2021.

[73] A. Meyer, T. Fritz, and G. C. Murphy, "Agile usability: Measuring user experience in agile development," *ACM Trans. Softw. Eng. Methodol*, vol. 32, no. 1, pp. 1–35, 2023.

[74] I. Jahnke, P. Bergström, E. Mårell-Olsson, L. Häll, and S. Kumar, "Digital didactical designs as research framework: iPad integration in Nordic schools," *Computers & Education*, vol. 113, pp. 1-15, 2017.

[75] E. García-López, L. Fernández-Sanz, J. Gómez-Pérez, and C. Pagés-Arévalo, "UX design in digital learning environments: Evidence-based practices," *Comput. Educ*, vol. 172, p. 104265, 2021. https://doi.org/10.1016/j.compedu.2021.104265

[76] K. Tanaka, J. R. Smith, L. Chen, M. P. Johnson, and A. B. Williams, "Critical factors in educational usability design," *Int. J. Hum. Comput. Stud*, vol. 144, p. 102495, 2020.

[77] M. Rodríguez, A. García-Holgado, D. Hernández-Leo, J. M. Dodero, and F. J. García-Peñalvo, "User participation in educational platform design," *IEEE Trans. Learn. Technol.*, vol. 15, no. 3, pp. 345–359, 2022. https://doi.org/10.1109/TLT.2022.3143226

[78] D. Ifenthaler, "Learning analytics for schools," *J. Educ. Technol. Syst*, vol. 49, no. 2, pp. 152–170, 2020. https://doi.org/10.1177/0047239520934018

[79] D. Ifenthaler, *Learning analytics for instructional design and support*. Switzerland AG: Springer Nature, 2021.

[80] M. J. Rodríguez-Triana, "Learning analytics for formative assessment," *Br. J. Educ. Technol*, vol. 54, no. 1, pp. 5–27, 2023.

[81] H. Zhang and L. Chen, "AI adoption in EdTech development: An analysis of GitHub repositories," *J. Educ. Technol. Syst*, vol. 52, no. 3, pp. 1–15, 2024.

[82] R. M. Hernández, M. González, and J. Díaz, "Software architecture patterns for educational technology platforms: A technical perspective," *IEEE Access*, vol. 10, 2022.

[83] A. Alam and G. Shi, "Technical-pedagogical considerations in developing AI-powered educational technologies," *J. Educ. Comput. Res*, vol. 16, no. 2, pp. 423–451, 2023. https://doi.org/10.1177/07356331221134325

[84] W. Li, R. Wang, and M. García, "Automated learning analytics platforms: Technical reliability vs. pedagogical adoption," *Comput. \& Educ*, vol. 215, pp. 1045–1060, 2024.

[85] S. Radović, M. Marić, and D. Passey, "Technology enhancing mathematics learning behaviours: Shifting learning goals from "producing the right answer" to "understanding how to address current and future mathematical challenges"," *Education and Information Technologies*, vol. 24, pp. 103-126, 2019.

[86] K. O'Dell, "Modern marginalia: Using digital annotation in the composition classroom," *Computers and Composition*, vol. 56, p. 102570, 2020. https://doi.org/10.1016/j.compcom.2020.102570