Federal University of Juiz de Fora
Institute of Exact Sciences
Bachelor's Degree in Computer Science

Architectual Stories:

A Card Game for Architectural Patterns Education

Maria Clara Ribeiro de Menezes

JUIZ DE FORA
JANUARY, 2026

Architectural Stories:
A Card Game for Architectural Patterns Education

Maria Clara Ribeiro de Menezes

Federal University of Juiz de Fora
Institute of Exact Sciences
Department of Computer Science
Bachelor's Degree in Computer Science

Advisor: Alessandreia Marta de Oliveira
Co-Advisor: Pedro Henrique Dias Valle

JUIZ DE FORA
JANUARY, 2026

ARCHITECTURAL STORIES:
A Card Game for Architectural Patterns Education

Maria Clara Ribeiro de Menezes

MONOGRAPH SUBMITTED TO THE FACULTY OF THE INSTITUTE OF EXACT SCIENCES OF THE FEDERAL UNIVERSITY OF JUIZ DE FORA, AS AN INTEGRAL PART OF THE NECESSARY REQUIREMENTS TO OBTAIN A BACHELOR'S DEGREE IN COMPUTER SCIENCE.

Accepted by:

Alessandreia Marta de Oliveira
Doctor in Computer Science - IC/UFF

Pedro Henrique Dias Valle
Doctor in Computer Science and Computational Mathematics - ICMC/USP

Marcela Sávia Picanço Pessoa
Doctor in Computer Science - UFAM

André Luiz de Oliveira
Doctor in Computer Science and Computational Mathematics - ICMC/USP

JUIZ DE FORA
JANUARY 09, 2026

To my parents, for their support and unconditional love.

Abstract

Software Architecture education presents significant pedagogical challenges, primarily due to the abstract nature of its concepts and the difficulty students face in connecting theory to practice. To understand this, we first conducted a Systematic Mapping Study between 2005 and 2025, analyzing 45 primary studies. The results revealed that while educational games exist, they often fail to address the specific skill of identifying architectural patterns from narrative-based problem descriptions. To bridge this gap, we propose Architectural Stories, a non-digital card game based on deductive reasoning mechanics. The game requires players to investigate Architectural situations, analyze symptoms, and deduce the appropriate architectural pattern to solve the enigma. To evaluate the proposed approach, we conducted a case study with 34 undergraduate students using three validated instruments: the Model for the Evaluation of Educational Game (MEEGA+), the Intrinsic Motivation Inventory (IMI), and the Instructional Materials Motivation Survey (IMMS). The results were highly positive, indicating that the game promotes strong social interaction, engagement, and intrinsic motivation. Furthermore, the evaluation confirmed that Architectural Stories is an effective instructional tool for developing diagnostic reasoning and reinforcing the understanding of architectural patterns in a collaborative environment.

Keywords: Software Architecture Education, Game-Based Learning, Architectural Patterns, Educational Games.

Resumo

O ensino de Arquitetura de Software apresenta desafios pedagógicos significativos, principalmente devido à natureza abstrata de seus conceitos e à dificuldade que os estudantes têm em conectar teoria e prática. Para enfrentar essa questão, foi realizado, primeiramente, um Mapeamento Sistemático da Literatura entre 2005 e 2025, com análise de 45 estudos primários. Os resultados revelaram que, embora existam jogos educacionais na área, frequentemente falham em abordar a habilidade específica de identificar padrões arquiteturais a partir de descrições narrativas de problemas. Para preencher essa lacuna, foi proposto o Architectural Stories, um jogo de cartas não digital baseado em mecânicas de raciocínio dedutivo. O jogo exige que os jogadores investiguem situações arquiteturais, analisem sintomas e deduzam o padrão arquitetural adequado para solucionar o enigma. Para avaliar a abordagem proposta, foi conduzido um estudo de caso com 34 estudantes de graduação, utilizando três instrumentos validados: *Model for the Evaluation of Educational Game (MEEGA+)*, *Intrinsic Motivation Inventory (IMI)* e *Instructional Materials Motivation Survey (IMMS)*. Os resultados foram altamente positivos, indicando que o jogo promove uma forte interação social, engajamento e motivação intrínseca. Além disso, a avaliação confirmou que o Architectural Stories é uma ferramenta instrucional eficaz para desenvolver o raciocínio diagnóstico e reforçar a compreensão de padrões arquiteturais em um ambiente colaborativo.

Palavras-chave: Ensino de Arquitetura de Software, Aprendizagem Baseada em Jogos, Padrões Arquiteturais, Jogos Educacionais.

Acknowledgments

First, I thank God for granting me the strength, health, and wisdom I needed throughout my academic journey, enabling me to overcome challenges and reach this significant milestone.

To my parents, Edna and Américo, I offer my deepest gratitude for all the love, support, and dedication they have shown throughout the years. Thank you for always believing in me, supporting my decisions, and doing everything in your power, even the impossible, so that I could achieve my goals. Without you, none of this would be possible.

I also thank my grandmother Aparecida and all my family members for their constant affection, encouragement, and support.

To Brian, my godson, for bringing light, joy, and life to my most difficult days.

To my friends, I am grateful for your presence at every moment, and for your understanding, companionship, and support along this path. A special thank you to the friends I made in college: thank you for making the days lighter and more fun, and for making this academic journey more pleasant. Without you, this path would certainly have been much harder.

I am very grateful to the Escola de Games UFJF project for all the learning and opportunities it provided, which contributed significantly to my professional growth. I also thank the Café das Minas initiative for welcoming me and including me in such an important and inspiring community.

To my professors, I am grateful for the knowledge and advice we shared, which significantly contributed to my academic and personal development. In particular, to my advisor, Professor Alessandreia Marta de Oliveira, I extend my sincere gratitude for all the support over the years, for your kindness, guidance, patience, and friendship. To my co-advisor, Professor Pedro Henrique Dias Valle, I leave my sincere gratitude for all the opportunities I had thanks to you, for your friendship, patience, and valuable lessons. Both of you were essential to the development and completion of this work.

Contents

List of Figures

List of Tables

List of Abbreviations

ADD	Attribute-Driven Design
AM	Architecture Master
ARCS	Attention, Relevance, Confidence, and Satisfaction
ATAM	Architecture Trade-off Analysis Method
CI/CD	Continuous Integration and Continuous Deployment
COMET	Concurrent Object Modeling and Architectural Design Method
GBL	Game-Based Learning
ICF	Informed Consent Form
IMI	Intrinsic Motivation Inventory
IMMS	Instructional Materials Motivation Survey
IoT	Internet of Things
IP	Insight Point
MEEGA+	Model for the Evaluation of Educational Games
MVC	Model-View-Controller
P2P	Peer-to-Peer
PBL	Project-Based Learning
PMA	Post-Mortem Analysis
RPG	Role-Playing Game
RQ	Research Question
SA	Software Architecture
SAE	Software Architecture Education
SMS	Systematic Mapping Study
SOA	Service-Oriented Architecture
SOL	SBC Open Library
UML	Unified Modeling Language

1 Introduction

1.1 Context and Problem

Software Architecture (SA) is characterized as the set of structures necessary to reason about a software system, comprising software elements and the relationships among them (BASS; CLEMENTS; KAZMAN, 2021). Consequently, SA serves as the critical link between design and requirements engineering, identifying the primary structural components of a system and how they are organized into communicating entities (SOMMERVILLE, 2018). Designing an SA capable of meeting system requirements is a crucial activity in the initial stages of software product development (OLIVEIRA et al., 2022). Therefore, architectural decisions directly impact both business objectives and the functional and quality requirements of the software (SOUSA; MARQUES, 2020).

A pivotal aspect of developing software with an architectural focus involves selecting an architectural style or pattern (BASS; CLEMENTS; KAZMAN, 2021). An architectural pattern can be described as a reusable description of a proven organizational structure, addressing recurring design problems that emerge within specific contexts (SOMMERVILLE, 2018). These solutions define the roles, responsibilities, and interactions of elements, often bundling multiple architectural tactics that inherently introduce trade-offs among quality attributes (BASS; CLEMENTS; KAZMAN, 2021).

Despite this, Software Architecture Education (SAE) is important for future IT professionals and presents substantial pedagogical challenges (GALSTER; ANGELOV, 2016; ARAÚJO et al., 2024). The abstract and imprecise nature of architectural concepts often impedes students from grasping the content meaningfully through traditional lecture-based instruction (OUH; IRAWAN, 2019; OLIVEIRA et al., 2022; CASTRO, 2023). Furthermore, lectures frequently fail to provide practical experience in decision-making and negotiation (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017). To understand architecture effectively, hands-on experience and active methodologies are required to complement traditional approaches (BOER; FARENHORST; VLIET, 2009;

DEURSEN et al., 2017). A specific challenge arises in training students to identify architectural patterns from narrative descriptions, a competency that demands technical proficiency and diagnostic reasoning that extends beyond the classroom environment (LAGO et al., 2019).

In this scenario, several strategies have been addressed, including project-based learning (VIDONI; MONTAGNA; VECCHIETTI, 2018), gamification, and educational tools (CARVALHO et al., 2025). In particular, educational games serve as effective instruments to promote engagement, critical thinking, and content assimilation playfully (GIACOBO, 2023; SOUZA et al., 2023; FEICHAS; SEABRA; SOUZA, 2021; LELIS, 2024). However, while the scientific literature shows several proposals, such as those focusing on design methods like Attribute-Driven Design (ADD) (CERVANTES et al., 2016), evaluation methods like the Architecture Trade-off Analysis Method (ATAM) (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017), or decision-making processes (LAGO et al., 2019), there is a notable deficiency in approaches specifically designed to train the skill of pattern identification from narrative contexts. Existing games predominantly offer predefined options or focus on method execution, rather than challenging the student to deduce the appropriate pattern solely from a description of symptoms.

1.2 Research Question

Considering the challenges inherent to SAE, particularly the difficulty students face in identifying and applying Architectural Patterns from narrative problem descriptions, this work seeks to answer the following research question (RQ):

> **RQ:** Can the use of a card game influence students' motivation to learn software architectural patterns?

1.3 Objectives

Motivated by the challenges students face in abstracting real-world scenarios into architectural patterns, the general objective of this work is to design, develop, and formally evaluate Architectural Stories, a non-digital educational card game aimed at supporting the SAE. Therefore, this work proposes a game-based learning approach to assist students in interpreting architectural problem narratives and mapping them to relevant SA patterns, thereby fostering diagnostic reasoning and active engagement.

To achieve the main objective, the following specific objectives were established:

- **Analyze the state of the art on SAE:** Examine existing game-based learning approaches in SA to identify gaps in the teaching and learning of architectural patterns;
- **Design and development of a collaborative instructional tool:** Define and develop the mechanics, components, and rules of a non-digital card game that integrates social interaction with core architectural concepts;
- **Evaluate the impact on student motivation:** Assess the effectiveness of the proposed game in terms of player experience, engagement, and perceived learning, employing validated instruments.

To achieve the established goals, we introduce and evaluate Architectural Stories, a non-digital card game designed to support the architectural patterns education. In contrast to initiatives focused on broad conceptual reinforcement, Architectural Stories concentrates directly on helping students interpret narratives and map them to patterns. The game employs a narrative deduction format inspired by Black Stories[1], wherein players collaboratively investigate Architectural situations until they reach a comprehensive solution. Through this structure, the game aims to foster practical understanding and logical reasoning in a clear and accessible format.

1.4 Methodology

To achieve the general objective, we adopted a structured research methodology organized into five distinct phases: (1) Identification of Existing Approaches, (2) Definition of Learning Objectives, (3) Game Design and Development, (4) Evaluation Planning and Execution, and (5) Data Collection and Analysis. These phases, summarized in Figure 1.1, are described below.

[1] https://tinyurl.com/mry3zr5h

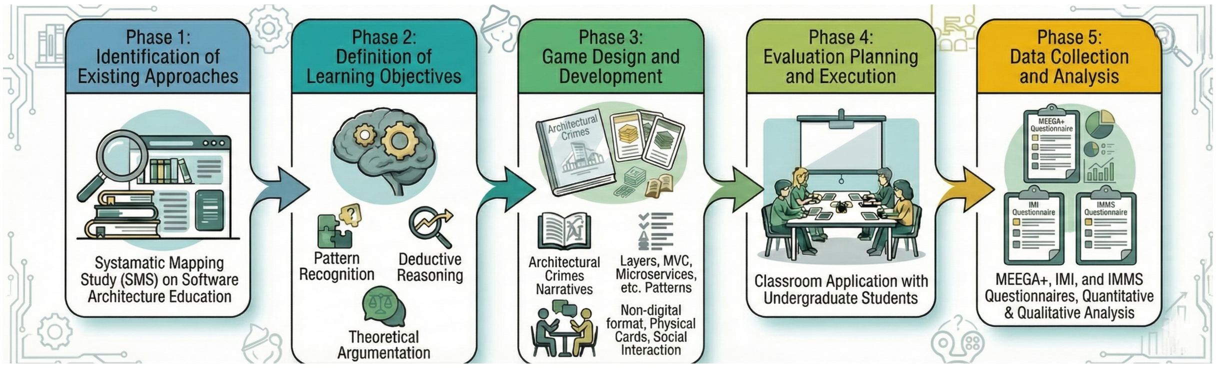


Figure 1.1: Research Methodology

Phase 1: Identification of Existing Approaches

We conducted a Systematic Mapping Study (SMS), as detailed in Chapter 3, on SAE to identify pedagogical strategies and game-based approaches currently available in the literature. The review process involved defining search strings, selecting digital libraries, and applying inclusion and exclusion criteria to select relevant studies.

While the review highlighted games that effectively addressed design methods, architectural evaluation, decision-making, and conceptual reinforcement, our results revealed a specific gap: none of the identified games explicitly supported the inference of architectural patterns from narrative problem descriptions. This finding motivated the development of a new approach focused on diagnostic reasoning.

Phase 2: Definition of Learning Objectives

Based on the identified gap, we defined specific learning goals for the game. The central purpose is to transition students from passive conceptual understanding to active diagnostic application. The first objective is to enhance pattern recognition. To achieve this, students must interpret open-ended narrative scenarios. They are challenged to identify specific architectural symptoms within these contexts.

The second objective is to foster deductive reasoning. The game cultivates the skill of linking a problem diagnosis to a specific architectural solution. This process relies on logical deduction to map symptoms to the correct pattern.

Finally, the design encourages theoretical argumentation. Students are required to articulate their choices using technical vocabulary. They must also justify their decisions through trade-off analysis.

Phase 3: Game Design and Development

This subsection describes the development of Architectural Stories. We crafted a set of Architectural situations. These are narratives that describe system failures, bottlenecks, or requirements violations without explicitly naming the underlying issue. The narratives were designed to be open-ended, requiring players to ask investigative questions to uncover the root cause.

We selected a comprehensive set of 19 architectural patterns to serve as the solutions. The selection criteria focused on patterns that represent fundamental structural styles and widely used distributed solutions in the software industry. The specific patterns included are Layers, Client-Server, Model-View-Controller (MVC), Pipe and Filter, Event-Driven Architecture, Service-Oriented Architecture (SOA), Publish-Subscribe, Microservices, Microkernel, Broker, Peer-to-Peer (P2P), Hexagonal Architecture, Saga Pattern, Adapter, Facade, Proxy, Mediator, Message Bus, and Blackboard.

Regarding the format, we deliberately chose a non-digital approach using physical cards for this initial version. This choice prioritizes face-to-face social interaction, negotiation, and oral argumentation. These soft skills are critical for software architects but are often harder to practice in digital environments. However, a digital version is currently under construction to facilitate future remote learning applications. Finally, the gameplay adopts a hybrid competitive and cooperative structure. This format was chosen to foster engagement through competition while simultaneously encouraging collaborative reasoning during the investigation phase.

Phase 4: Evaluation Planning and Execution

To evaluate the game's educational effectiveness, usability, and motivational aspects, we designed an evaluation study conducted in a classroom setting.

The evaluation was planned as a case study within an undergraduate Software Engineering course. To comprehensively assess different dimensions of the educational game, three validated instruments were selected.

First, the MEEGA+ (Model for the Evaluation of Educational Games) (PETRI; WANGENHEIM; BORGATTO, 2017) was employed to assess the perceived quality of the game, encompassing aspects related to usability, player experience, and perceived learning. Second, the Intrinsic Motivation Inventory (IMI) (RYAN; KOESTNER; DECI, 1991) was employed to measure students' intrinsic motivation and experience during gameplay, focusing on dimensions such as interest/enjoyment, perceived competence, and effort. Finally, the Instructional Materials Motivation Survey (IMMS) (KELLER, 2009) was administered to assess the motivational aspects of the game based on the ARCS model, encompassing attention, relevance, confidence, and satisfaction.

The combination of these instruments enabled a multidimensional evaluation of the game, addressing engagement, motivation, usability, and perceived educational value.

The application took place in a classroom setting with 34 undergraduate students. Upon arrival, the students organized themselves into groups of three to five members. Each group received the game materials (mystery card decks, pattern summary sheets, and scoreboards) and was instructed to learn the rules exclusively from the provided manuals.

During the activity, the instructor remained available only for occasional clarifications, deliberately avoiding interference in the groups' reasoning and interactions. This approach was intentional, as one of the goals was to evaluate whether students could understand and engage with the game solely by relying on the instruction manual.

Phase 5: Data Collection and Analysis

Immediately after the gameplay session, data collection was performed. Students answered the MEEGA+, IMI, and IMMS questionnaires individually and anonymously. The instruments included quantitative items measured on Likert scales, as well as open-ended questions to capture qualitative feedback regarding the game's strengths and areas for improvement.

Quantitative data were analyzed using descriptive statistics (mean and standard deviation) to assess levels of acceptance, motivation, engagement, and perceived quality. Qualitative comments were read in detail and interpreted in conjunction with the quantitative results, allowing the identification of consistent perceptions related to the game's mechanics, engagement, motivational impact, and learning value.

1.5 Organization

This document is structured as follows. Chapter 2 presents the theoretical foundations that support this research, including concepts related to SA, educational games, and learning theories relevant to computing education. Chapter 3 discusses the related work, presenting an SMS and the primary studies identified through this process, highlighting existing approaches, research gaps, and trends in the SAE. Chapter 4 introduces the Architectural Stories Game, detailing its educational objectives, game design, mechanics, and the architectural decisions adopted during its development. Chapter 5 presents the evaluation process and discusses the obtained results, including the empirical study design, evaluation instruments, data analysis, and findings related to usability, motivation, engagement, and perceived learning. Finally, Chapter 6 concludes this work by summarizing the main contributions, discussing limitations, and outlining directions for future research.

2 Theoretic Foundations

This chapter presents the theoretical foundations that underpin this research, providing the conceptual background necessary to understand its scope and objectives. It is organized into three main sections. Section 2.1 discusses the fundamentals of Software Architecture, highlighting its definition and relevance in the software development lifecycle. Section 2.2 addresses Architectural Patterns, emphasizing their importance and the inherent challenges involved in teaching and learning them. Finally, Section 2.3 explores the field of Educational Games, defining their main characteristics and elements, and examining their role as a pedagogical strategy to mitigate the difficulties associated with learning complex technical concepts.

2.1 Software Architecture

Software Architecture is a core discipline in Software Engineering, serving as the blueprint for system construction and evolution. According to Bass, Clements and Kazman (2021), SA consists of the set of structures necessary to reason about the system, which comprises software elements, the relations among them, and the properties of both. It acts as a bridge between requirements engineering and the implementation phase, translating business goals and functional requirements into a technical structure (SOMMERVILLE, 2018).

The importance of SA lies in its impact on the quality attributes of the system, such as performance, security, modifiability, and reliability. Architectural decisions are typically made early in the development lifecycle and are notoriously difficult and costly to change at later stages (OLIVEIRA et al., 2022). Therefore, a well-defined architecture is essential for project success, as it facilitates communication among stakeholders, enables the reuse of components, and allows for the analysis of the system behavior before it is fully implemented (BASS; CLEMENTS; KAZMAN, 2021).

Despite its importance, SA is considered one of the most difficult subjects to teach and learn in Computer Science curricula (GALSTER; ANGELOV, 2016). The abstract nature of the concepts, combined with the need for high-level abstraction skills and the ability to balance trade-offs, poses significant challenges for novice students (CASTRO, 2023). Unlike code-level programming, which provides immediate feedback through compilation, architectural design requires diagnostic reasoning and long-term vision, skills that are difficult to acquire through traditional theoretical lectures alone (OUH; IRAWAN, 2019).

2.2 Architectural Patterns

A central concept in SA is the Architectural Pattern. A pattern describes a proven solution to a recurring design problem that arises in specific contexts (SOMMERVILLE, 2018). These patterns capture static and dynamic structures and collaborations of components that have been successful in previous systems.

Architectural patterns differ from design patterns in terms of scope and level of abstraction: while architectural patterns address high-level system organization, defining the overall structure of the software and the relationships among its major components, design patterns focus on localized design problems, providing reusable solutions at the level of classes, objects, and their interactions (GAMMA, 1995).

Familiarity with architectural patterns is crucial for software architects because patterns provide a shared vocabulary and a standardized approach to addressing quality requirements. By encapsulating architectural knowledge accumulated through prior experience, patterns support architects in reasoning about design alternatives and in making informed architectural design decisions (BASS; CLEMENTS; KAZMAN, 2021). As such, architectural patterns play a fundamental role in early development phases, when key decisions have long-lasting impact on system evolution, maintainability, and overall quality (TYREE; AKERMAN, 2005; BASS; CLEMENTS; KAZMAN, 2021).

Some of the most widely known patterns include Model-View-Controller (MVC), Layered Architecture, Client-Server, Microservices, and Pipe and Filter, among others (BASS; CLEMENTS; KAZMAN, 2021). Each of these patterns promotes specific quality attributes while potentially inhibiting others. For example, a Layered architecture may enhance modifiability and separation of concerns, but it can negatively impact performance due to the overhead of data traversing multiple layers. Similarly, Microservices architectures support scalability and independent deployment, at the cost of increased system complexity and operational overhead.

Architectural patterns are therefore closely related to the concept of architectural trade-offs. Selecting a pattern rarely leads to the simultaneous optimization of all quality attributes; instead, architects must prioritize certain qualities over others based on system goals and constraints (BASS; CLEMENTS; KAZMAN, 2021). This trade-off-oriented nature of architectural design reinforces the importance of understanding not only the structural aspects of patterns, but also their consequences on quality attributes such as performance, reliability, and security (SOMMERVILLE, 2018).

Despite their importance, learning to identify and select the appropriate architectural pattern for a given scenario is a complex cognitive task. It involves not only memorizing the structural characteristics of patterns, but, more importantly, understanding the problem context that necessitates a specific architectural solution (XAVIER; WERNER; TRAVASSOS, 2002). Students are often required to interpret informal or narrative descriptions of requirements, quality concerns, or system failures and map them to abstract architectural concepts, which demands higher-order reasoning skills.

Empirical studies indicate that students frequently struggle to correctly diagnose architectural problems and associate them with suitable patterns (LAGO et al., 2019). This difficulty is further exacerbated by the abstract nature of architectural patterns and by the limited exposure of students to real-world systems during their training (GALSTER; ANGELOV, 2016). As a result, learners may develop a superficial understanding of patterns, recognizing their names and structures without fully grasping the contextual factors that justify their application.

Moreover, traditional teaching approaches for SA, often centered on lectures, static diagrams, and textbook examples, tend to emphasize theoretical knowledge over practical decision-making skills (RAZMOV, 2007). Such approaches are often insufficient to convey the dynamic and context-dependent nature of architectural reasoning, which involves balancing quality attributes and trade-offs (BASS; CLEMENTS; KAZ-

MAN, 2021). As a result, students often face difficulties when applying architectural patterns to realistic problem scenarios (LAGO et al., 2019). These limitations highlight the need for instructional strategies that actively engage learners in contextualized architectural problem-solving and decision-making processes (FREEMAN et al., 2014).

In this context, active learning methodologies have been increasingly explored as alternatives to traditional instruction in software engineering education (LELIS, 2024). Approaches that leverage simulation and interaction allow students to experience the consequences of architectural decisions in a controlled learning environment (KAZMAN et al., 1998). Experiential learning theories suggest that placing learners in situations that require analysis, diagnosis, and justification promotes deeper understanding and higher-order cognitive skills (SINGH; RAO, 2024). Consequently, such methodologies aim to bridge the gap between theoretical knowledge of architectural patterns and their practical application in real-world contexts (WANG; WU, 2011).

2.3 Educational Games

To address the challenges of teaching complex and abstract topics, Game-Based Learning (GBL) has emerged as a promising approach to education. Educational games are activities that have clear learning objectives and leverage the entertainment power of games to achieve them. Unlike pure entertainment games, educational games are designed to balance the fun factor with specific instructional goals (WANG; WU, 2011).

2.3.1 Definitions and Game Elements

A game can be defined as a structured system in which players engage in an artificial conflict governed by explicit rules, with the objective of producing a quantifiable outcome (TEKINBAS; ZIMMERMAN, 2003). In educational contexts, games are particularly valuable because they create safe environments in which failure is not only permitted but expected, thereby encouraging experimentation, reflection, and active participation (GEE, 2003). This characteristic makes games especially suitable for learning complex and abstract subjects, such as SA, where understanding often emerges through trial, error, and iterative reasoning (SINGH; RAO, 2024).

Regardless of whether they are digital or non-digital, games are composed of fundamental elements that shape player engagement and learning experiences (DETERDING et al., 2011). Game mechanics define the formal structure of the game, specifying the rules, actions, constraints, and procedures that guide player behavior (ROBIN; MARC; ROBERT, 2004). As players interact with these mechanics, game dynamics emerge, reflecting the run-time behavior of the system and manifesting as patterns such as collaboration, competition, negotiation, or strategic decision-making (ROBIN; MARC; ROBERT, 2004). In parallel, the aesthetic and narrative components of a game contribute to the emotional and cognitive engagement of players by providing context, meaning, and immersion, which can enhance motivation and sustain attention over time (ROBIN; MARC; ROBERT, 2004). Together, these elements are reinforced by clearly defined goals and continuous feedback, enabling players to assess the consequences of their actions, correct misunderstandings, and consolidate learning through repeated interaction (GALSTER; ANGELOV, 2016).

Digital educational games have been widely explored within Software Engineering education, addressing topics that range from project management and requirements engineering to programming and software design (CERVANTES et al., 2016). Their popularity can be attributed to several advantages, including the ability to provide automated and immediate feedback, simulate complex systems and processes, and visually represent abstract concepts that are otherwise difficult to grasp through traditional instructional methods (WU; WANG, 2012). In addition, digital games can support scalability and learning analytics, enabling instructors to monitor learner progress and adapt instructional strategies based on collected data (CONNOLLY et al., 2012).

Nevertheless, non-digital games, such as card games and board games, also play a significant and complementary role in educational settings (WOUTERS et al., 2013). These games emphasize direct social interaction, face-to-face communication, and collective decision-making, fostering skills such as negotiation, argumentation, and collaborative problem solving. Such skills are particularly relevant in the domain of SA, where architectural decisions are rarely made in isolation and often require discussion, justification, and consensus among multiple stakeholders (BASS; CLEMENTS; KAZMAN, 2021). As a result, non-digital games can offer pedagogical benefits that are difficult to replicate in purely digital environments, reinforcing their relevance in Software Engineering education (SOUSA; MARQUES, 2020).

2.3.2 Games in Software Architecture Education

The use of games in SAE seeks to shift learning from a predominantly passive, instructor-centered approach toward a more active and student-centered experience. Traditional instructional strategies often emphasize theoretical exposition and static representations, which may be insufficient to capture the complexity and trade-offs inherent in architectural decision making (GALSTER; ANGELOV, 2016; BASS; CLEMENTS; KAZMAN, 2021). In contrast, educational games enable the simulation of architectural decision-making processes, allowing students to explore alternative solutions, reason about quality attributes, and observe the consequences of their choices within a controlled and low-risk environment (CERVANTES et al., 2016; BOER et al., 2019).

Several studies have shown that game-based approaches can effectively support the development of architectural reasoning skills by embedding learners in realistic problem scenarios that require analysis, justification, and negotiation of design decisions (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017; LAGO et al., 2019). By incorporating elements such as competition, collaboration, and role-playing, games can also foster higher levels of engagement and motivation when compared to conventional exercises or assignments (WANG; WU, 2011; SOUZA et al., 2023). These characteristics are particularly relevant in SAE, where learning outcomes depend not only on technical knowledge but also on communication, argumentation, and consensus building among stakeholders (TYREE; AKERMAN, 2005; DEURSEN et al., 2017).

Despite the growing body of work proposing and applying games in SA courses, a recurring challenge remains in systematically evaluating how effectively these tools support specific learning objectives and competencies. In particular, skills such as diagnostic reasoning, architectural pattern identification, and trade-off analysis require careful assessment to determine whether game-based interventions lead to meaningful learning gains

(OLIVEIRA et al., 2022; YÉPEZ et al., 2024). Consequently, the evaluation of educational games commonly relies on validated instruments that address both pedagogical quality and learner experience.

Among these instruments, the MEEGA+ has been widely adopted to assess the quality of educational games in terms of usability, player experience, and perceived learning. To complement usability and experience-oriented evaluations, motivation-focused instruments are frequently employed. The IMI is used to measure learners' subjective experiences related to interest, enjoyment, and perceived competence during the activity. Similarly, the IMMS, grounded in Keller's ARCS model, evaluates the motivational impact of instructional artifacts by examining dimensions such as attention, relevance, confidence, and satisfaction. Together, these instruments provide a structured and multidimensional perspective on the effectiveness of games as pedagogical tools in SAE.

2.4 Final Considerations

This chapter established the theoretical foundations of SA and the specific challenges related to teaching Architectural Patterns due to their abstract nature. It also highlighted Game-Based Learning as a viable strategy to bridge the gap between theory and practice. To better understand the current landscape of educational initiatives in this domain, the following chapter presents an SMS of the literature, which investigates existing approaches, tools, and games used to teach SA over the last two decades.

3 Related Work

This chapter presents an SMS on SAE, serving as the empirical foundation for this research. The study detailed herein is based on our previously published work (MENEZES; VALLE; OLIVEIRA, 2026b), which provides a comprehensive synthesis of two decades of research in the field. The remainder of this chapter is organized as follows: Section 3.1 contextualizes the importance of SAE and the role of an SMS; Section 3.2 details the research protocol, including questions, search string, and selection criteria; Section 3.3 presents the synthesis of findings and the discussion of the research questions; Section 3.4 outlines future directions for the area; and, finally, Section 3.5 and Section 3.6 discuss the study's limitations and final considerations.

3.1 Contextualization

SA is pivotal to the success of software systems, yet teaching it effectively remains an open problem in software engineering education. Its abstraction, cognitive demands, and the need to integrate technical and social skills pose barriers for students and instructors. To address this, we conducted a systematic mapping study of 45 primary studies published between 2005 and 2025, providing the most comprehensive overview of SAE to date. Our analysis reveals that curricula still emphasize quality attributes, architectural patterns, documentation, and design processes, while active learning strategies, such as Project-Based Learning (PBL), gamification, and flipped classrooms, show a promising yet underexplored impact. At the same time, critical gaps persist: bridging abstraction with practice, designing for heterogeneous cohorts, and aligning educational practices with industry demands remain challenges. By consolidating two decades of research, this study presents updated evidence, identifies underexplored areas, and outlines directions for enhancing the effectiveness, scalability, and relevance of SAE.

A systematic mapping study (SMS) is a secondary study that aims to identify and classify content related to a specific research topic. An SMS investigates primary studies to identify available evidence and highlight knowledge gaps. In this way, the results provide insights into existing research gaps in the area, which may suggest future research directions and offer guidance on how to properly position new research activities. Hence, systematic mappings aim to provide an overview of a topic and to identify whether there are subtopics in which further primary studies are needed (SCANNAVINO et al., 2017).

The primary objective of this systematic mapping study is to identify, analyze, and synthesize the initiatives (approaches, methods, tools, and topics) currently employed in the SAE. This study aims to deepen the understanding of how SA is taught across diverse educational contexts, mapping existing practices, highlighting knowledge gaps, and identifying emerging trends, thereby offering a solid foundation for improving curricula and practices in the field. Therefore, the main contribution of this SMS is to provide a comprehensive and up-to-date synthesis of two decades of research (2005–2025) on SAE. Unlike previous secondary studies, this study expands its temporal coverage, integrates the latest initiatives, and focuses explicitly on practices across diverse educational contexts. By consolidating fragmented evidence, uncovering persistent gaps, and outlining actionable directions, our work establishes a foundation for advancing curricula, methods, and tools that help prepare students for both academic learning and real-world professional demands. We believe that the content of this work not only updates the state of the art but also helps shape the research agenda for the next decade concerning SAE.

In this context, this chapter plays a dual role. First, it reports the Systematic Mapping Study conducted to investigate SAE. Second, it presents and discusses the related work identified through this mapping. The 45 primary studies selected and analyzed in the SMS constitute the set of related works considered in this research. Rather than relying on an ad hoc selection, the related work is grounded in a rigorous and transparent secondary study process, ensuring comprehensive coverage and reducing selection bias.

3.2 Research Method

This section describes the methodological approach adopted in this study, detailing the steps used to plan, conduct, and report the SMS. The objective is to provide a clear and transparent account of the procedures, ensuring reproducibility and research rigor. The following subsections present the RQ, the search string, the inclusion and exclusion criteria, the databases, and the study selection process.

3.2.1 Research Questions (RQs)

The RQ were formulated based on the objectives defined for this SMS. These questions guide both the selection and the critical analysis of the identified studies, ensuring that the investigation remains aligned with the established scope:

- **RQ1:** What topics have been taught in the field of Software Architecture?
- **RQ2:** What types of approaches have been adopted to teach Software Architecture?
- **RQ3:** What evidence exists regarding the effectiveness of these approaches in the teaching-learning process of students?
- **RQ4:** What are the main challenges in Software Architecture Education?

These questions were designed to guide the extraction of relevant information from the primary studies, enabling a consistent and detailed analysis of the themes addressed in the literature.

3.2.2 Search String

The definition of the search string for this SMS was based on a careful analysis of the study's objective and the most representative keywords for SAE. To ensure maximum coverage of relevant studies, the string was iteratively calibrated through tests with different combinations of terms, Boolean operators, and wildcard expressions. This process enabled the identification of expressions capable of retrieving key studies in the field, while minimizing the inclusion of irrelevant publications.

The search string was structured to capture three main dimensions: (i) concepts related to SA, (ii) educational actions or learning processes, and (iii) innovative methods or tools applied to teaching. In addition, specific exclusions were defined to avoid results outside the scope of this study, such as research on urban architecture or computer architecture, which use similar terminology but are unrelated to SAE.

For the first dimension, referring to SA concepts, terms such as *SA*, *system architecture*, *architectural design*, *architecture design*, *architectural decisions*, *software structure*, and *architectural views* were selected. These terms were chosen to encompass both theoretical approaches and practical applications, including architectural decisions and the structuring of complex systems. The use of the OR operator between these terms ensures that any study containing at least one of the core concepts is retrieved, thereby broadening coverage.

The second dimension, related to educational actions, included terms such as *educat**, *learn**, and *teach**. The use of truncation (*) allows coverage of linguistic variations and different verb conjugations, ensuring that studies focusing on education, teaching, learning, or training are included regardless of minor terminological differences. Again, the OR operator was applied among these terms to capture all publications addressing teaching or learning.

The third dimension covered the use of innovative methods, tools, or strategies applied to the teaching of SA. This included terms such as *game*, *tool*, *gamification*, *simulation*, *flipped classroom*, PBL, *game-based learning*, and *course*. These terms reflect the diversity of approaches reported in the literature, ranging from educational games and simulations to computational tools and active teaching methodologies such as PBL and the Flipped Classroom. Once more, the OR operator was used among these terms to maximize the retrieval of relevant studies.

To avoid irrelevant results, the AND NOT operator was applied to exclude publications related to *urban* and *computer architecture*, which are not within the scope of this SMS. This exclusion step was essential to prevent bias and ensure that the retrieved sample of studies remained consistent with the study's focus. The final search string applied to the selected databases was defined as follows:

> (software architecture OR "system architecture" OR "architectural design" OR "architecture design"
> OR "architectural decisions" OR "software structure" OR "architectural views")

AND (educat* OR learn* OR teach*)

AND (game OR tool OR gamification OR simulation OR "flipped classroom"

OR "project-based learning" OR "game-based learning" OR course)

AND NOT (urban OR "computer architecture")

This final formulation demonstrates the intention to retrieve studies relevant to SAE, considering both practical and innovative approaches, and shows that the strategy was calibrated to strike a balance between breadth and precision. The string meets the objectives of the SMS, enabling the identification of significant and contemporary studies in the field while ensuring the exclusion of unrelated publications.

3.2.3 Control Studies

To validate the effectiveness of the search string, three studies considered essential to the field were selected as *control studies*. These studies represent key contributions to SAE, focusing on tools and educational games, and were used as verification criteria to ensure that the search strategy was able to retrieve them. The control studies chosen are in Table 3.1.

Table 3.1: Control studies used to validate the search string

ID	Reference
1	Urrego, Juan Sebastián, and Dario Correal. "Archinotes: A tool for assisting SA courses." 26th International Conference on Software Engineering Education and Training (CSEE&T). IEEE, 2013 (URREGO; CORREAL, 2013).
2	Cervantes, Humberto, Serge Haziyev, Olha Hrytsay, and Rick Kazman. "Smart decisions: an architectural design game." Proceedings of the 38th International Conference on Software Engineering Companion, 2016 (CERVANTES et al., 2016).
3	Wang, Alf Inge, and Bian Wu. "Using game development to teach SA." International Journal of Computer Games Technology 2011: 920873 (WANG; WU, 2011).

The inclusion of these papers in the search results demonstrated that the constructed string is robust, comprehensive, and precise, ensuring that central studies in the field are not omitted. Validation through control studies is a recommended methodological procedure in SMS, providing additional reliability to the process of term and operator selection.

3.2.4 Inclusion and Exclusion Criteria

Carefully defined inclusion and exclusion criteria were established in line with the objectives of this secondary study, ensuring that only relevant and adequately rigorous primary studies were considered in the analysis. The strict application of these criteria aimed to ensure the consistency of the selection process and the validity of the results obtained. The following inclusion criterion was defined for the selection of studies:

- **IC1:** The study reports an experience, resource, or approach related to SAE.

The following exclusion criteria were adopted to refine the set of selected studies:

- **EC1:** The study is not written in Portuguese or English;
- **EC2:** The study is not available;
- **EC3:** The study is a book or grey literature;
- **EC4:** The study is a short paper;
- **EC5:** The study is duplicated.

The adoption of these criteria ensures greater methodological rigor in the selection process, allowing the analyzed studies to effectively contribute to the understanding of practices, challenges, and trends related to SAE.

3.2.5 Digital Libraries

For the application of the search string, digital libraries widely recognized for the relevance and breadth of their collections of scientific publications in the fields of Software Engineering and Information Technology Education were selected. The choice of these digital libraries is justified by their representativeness and suitability for the study's objective, enabling the identification of primary studies relevant to the investigated topic. The digital libraries that were considered are Engineering Village[2], IEEE Xplore[3], Scopus[4], and

SOL (SBC Open Library)[5]. In the SOL database, the search was conducted manually, since the platform does not fully support the application of the predefined search string.

3.2.6 Study Selection

The selection of studies followed the PRISMA 2020 protocol, ensuring transparency, rigor, and reproducibility throughout the entire process. Figure 3.1 summarizes the main steps of the systematic mapping study (SMS), illustrating the progression from the initial identification of records to the final inclusion of primary studies. The selection of studies was conducted through a carefully systematic process, designed to guarantee both the relevance and the methodological quality of the publications included in the SMS.

As an initial step, all duplicate records identified across various digital libraries and databases were meticulously removed, thereby preventing overrepresentation and ensuring the accuracy of the dataset. This step also contributed to streamlining the subsequent screening phases, reducing the workload and focusing the analysis on unique studies.

The remaining studies were then processed through three well-defined phases. In the first phase, a preliminary screening was conducted based on titles and abstracts, guided by predefined inclusion and exclusion criteria. Studies that met these criteria advanced to the second phase, which involved a more detailed analysis of abstracts, introductions, and conclusions, allowing for the confirmation of relevance and the exclusion of studies that did not sufficiently align with the research objectives.

In the final phase, the selected studies were thoroughly reviewed, enabling data extraction, methodology identification, and evaluation of key findings. Only studies that effectively addressed the RQ and objectives were retained. The search string applied across the selected databases initially returned a total of 5,076 publications. The distribution of these results by digital library is presented in Figure 3.2, providing an overview of the volume of studies retrieved at the outset of the selection process. To accommodate the considerable variation across databases, a broken axis was applied. This technique

[2]https://www.engineeringvillage.com/home.url
[3]https://ieeexplore.ieee.org/Xplore/home.jsp
[4]https://www.scopus.com/home.uri
[5]https://sol.sbc.org.br/busca/

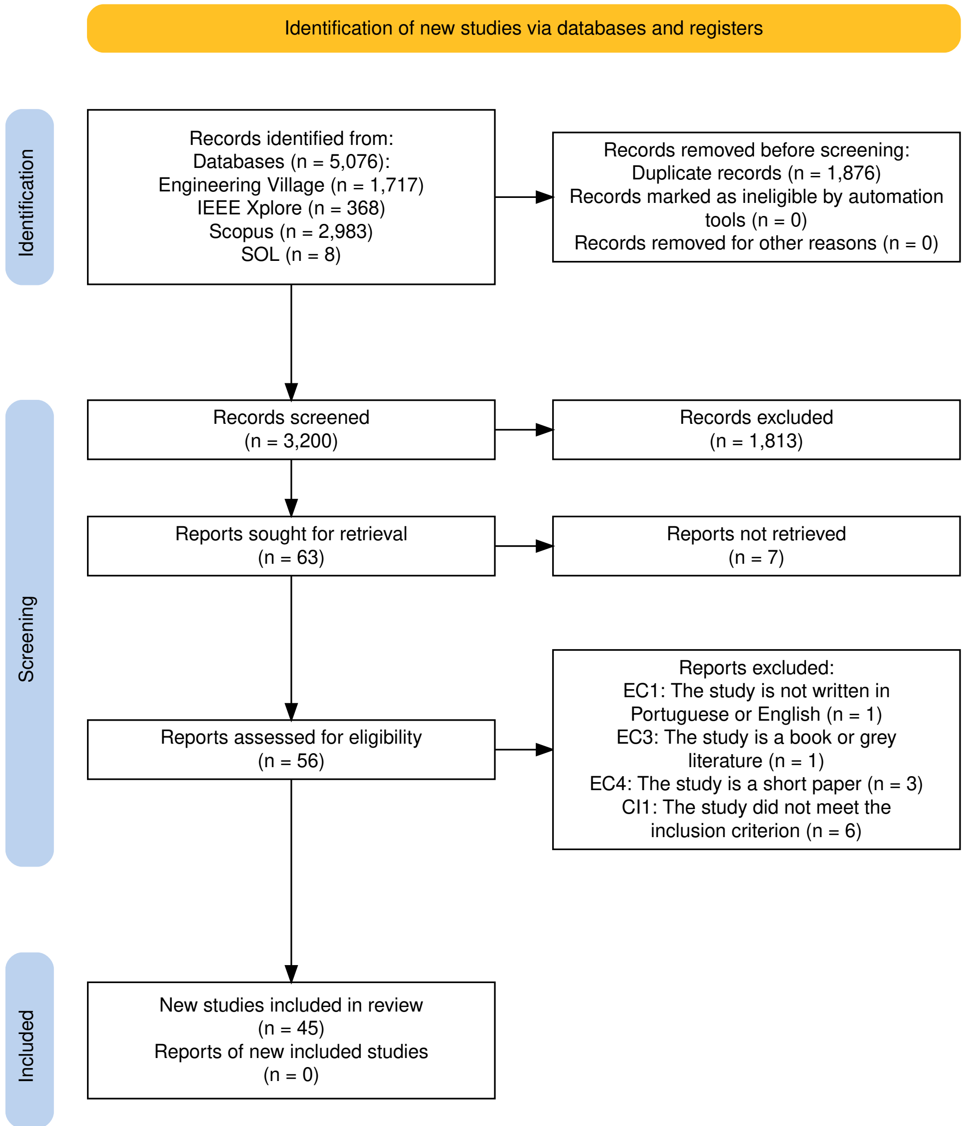


Figure 3.1: PRISMA flow diagram for the SMS.

minimizes the disproportionate space that would result from extreme values, such as the large gap between the smallest and largest counts, while still preserving a clear comparison of the relative contributions of each database.

After removing duplicates, the total comprised 3,200 unique studies. In the first screening (titles and abstracts), 1,813 were excluded, and 63 were deemed relevant, advancing to the next stage. In the second phase, involving a more comprehensive assessment of abstracts, introductions, and conclusions, a further 18 studies were excluded. The reasons for exclusion in this stage were diverse: one study was discarded because it was not written in either Portuguese or English (EC1); seven were excluded due to unavailability of the full text (EC2); one was identified as a book or gray literature rather than a peer-reviewed article (EC3); three were excluded because they were too short to be considered

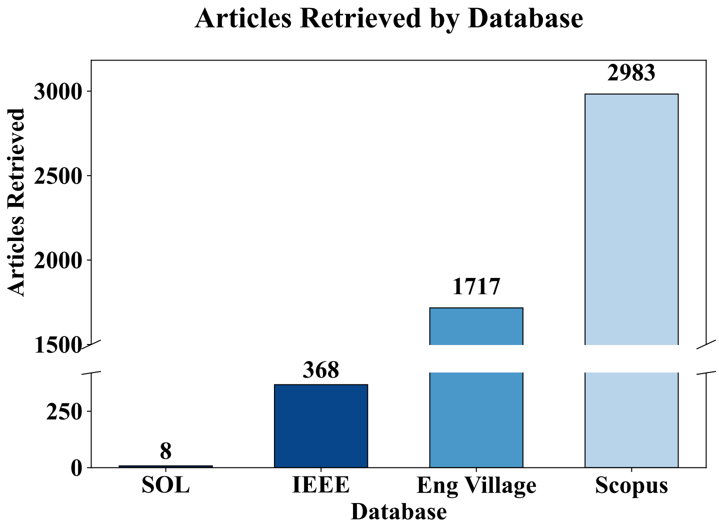


Figure 3.2: Studies retrieved per digital library

for analysis (EC4); and six were rejected for not fulfilling the inclusion criterion IC1.

As a result of this rigorous and systematic selection procedure, a final set of 45 primary studies was identified, all of which fully adhered to the predefined inclusion criteria. Collectively, these studies represent a broad spectrum of approaches, strategies, and empirical experiences related to SAE in various educational contexts. Table 3.2 enumerates the complete set of included studies, thereby offering a detailed basis for the subsequent stages of analysis and discussion. This transparent protocol ensures that the 45 primary studies analyzed represent a reliable and comprehensive evidence base for answering our research questions.

Table 3.2: Analyzed primary studies in this SMS.

ID	Title	Ref.
S1	A Collaborative approach to teaching software architecture	(DEURSEN et al., 2017)
S2	A community of learners approach to software architecture education	(BOER; FARENHORST; VLIET, 2009)
S3	Adapting cooperative learning to teach software architecture in multiple role-teams	(CHENOWETH; ARDIS; DUGAS, 2007)
S4	An approach to software architecting in agile software development projects in education	(ANGELOV; BEER, 2015)
S5	Applying case-based learning for a postgraduate software architecture course	(OUH; IRAWAN, 2019)
S6	Archinotes: A tool for assisting software architecture courses	(URREGO; CORREAL, 2013)
S7	ATAM-RPG: A role-playing game to teach architecture trade-off analysis method (ATAM)	(MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017)
S8	Avaliação e Melhoria da Experiência do Jogador em um Jogo para Ensino de Arquitetura de Software	(LIMA; MARQUES, 2024)

Continuation of Table 3.2

ID	Title	Ref.
S9	Comparison of learning software architecture by developing social applications versus games on the android platform	(WU; WANG, 2012)
S10	DecidArch V2: An Improved Game to Teach Architecture Design Decision Making	(BOER et al., 2019)
S11	Decidarch: Playing cards as software architects	(LAGO et al., 2019)
S12	DECORA: Um Sistema de Apoio ao Ensino de Decisões de Projetos Arquiteturais	(CARVALHO et al., 2025)
S13	Did our Course Design on Software Architecture meet our Student's Learning Expectations?	(OUH; GAN; IRAWAN, 2020)
S14	Educational approach to an experiment in a software architecture course	(WANG; ARISHOLM; JACCHERI, 2007)
S15	Exploring Experiential Learning Model and Risk Management Process for an Undergraduate Software Architecture Course	(LIEH; IRAWAN, 2018a)
S16	Exploring game architecture best-practices with classic space invaders	(KEENAN; STEELE, 2011)
S17	Extending Google Android's application as an educational tool	(WU et al., 2010)
S18	Extensive evaluation of using a game project in a software architecture course	(WANG, 2011)
S19	Flipped Classroom Applied to Software Architecture Teaching	(GONÇALVES et al., 2020)
S20	How Software Architects Learn: A pilot study of their learning style in Kolb's Learning Styles Inventory	(HIDALGO; ASTUDILLO; CASTRO, 2023)
S21	LEARN 2.0: Evolução de um jogo de tabuleiro para o ensino de Arquitetura de Software	(LELIS, 2024)
S22	Making the comprehension of software architecture attractive	(RODRIGUES; WERNER, 2011)
S23	Modeling in agile project courses	(ALPEROWITZ et al., 2017)
S24	Patterns and traceability in teaching software architecture	(GAST, 2008)
S25	Pedagogia Sistêmica: experiência do UniSENAI Campus Joinville no ensino de Arquitetura e Desenvolvimento de Software	(SOUZA; FREITAS; RENGEL, 2022)
S26	Post-mortem analysis of student game projects in a software architecture course	(WANG, 2009)
S27	Project and team-based strategies for teaching software architecture	(VIDONI; MONTAGNA; VECCHIETTI, 2018)
S28	Requirements and architecture modeling in software engineering courses	(GRBAC; CAR; VUKOVIĆ, 2015)
S29	Role-playing software architecture styles	(CASTRO, 2023)
S30	Scrum as a method of teaching software architecture	(WEDEMANN, 2018)
S31	Smart decisions: An architectural design game	(CERVANTES et al., 2016)
S32	Software engineering education toolkit for embedded software architecture design methodology using robotic systems	(KIM et al., 2008)
S33	Teaching a course on software architecture	(LAGO; VLIET, 2005)
S34	Teaching adult learners on software architecture design skills	(LIEH; IRAWAN, 2018b)
S35	Teaching data structures and software architecture while constructing curriculum platform	(CAO; CAO, 2011)
S36	Teaching Distributed Software Architecture by Building an Industrial Level E-Commerce Application	(WEI et al., 2020)

Continuation of Table 3.2

ID	Title	Ref.
S37	Teaching software architecture design	(MANNISTO; SAVOLAINEN; MYLLARNIEMI, 2008)
S38	Teaching Software Architecture to Undergraduate Students: An Experience Report	(RUPAKHETI; CHENOWETH, 2015)
S39	Todo Esforço Será Recompensado: Gamificação no Ensino de Arquitetura de Software com o Uso de Badges	(ARAÚJO et al., 2024)
S40	Toward Creating Software Architects Using Mobile Project-Based Learning Model (Mobile-PBL) for Teaching Software Architecture	(AL-QORA'N; JAWARNEH; NGANJI, 2023)
S41	Using an architecture reasoning tool to teach software architecture	(MCGREGOR et al., 2007)
S42	Using game development to teach software architecture	(WANG; WU, 2011)
S43	Using Post-Mortem Analysis to evaluate software architecture student projects	(WANG; STALHANE, 2005)
S44	What makes teaching software architecture difficult?	(GALSTER; ANGELOV, 2016)
S45	XQUEST used in software architecture education	(WU et al., 2009)

The word cloud presented in Figure 3.3 illustrates the terms that occur most frequently in the abstracts of the selected studies, providing a synthetic yet comprehensive overview of the thematic emphases found in the literature on SAE.

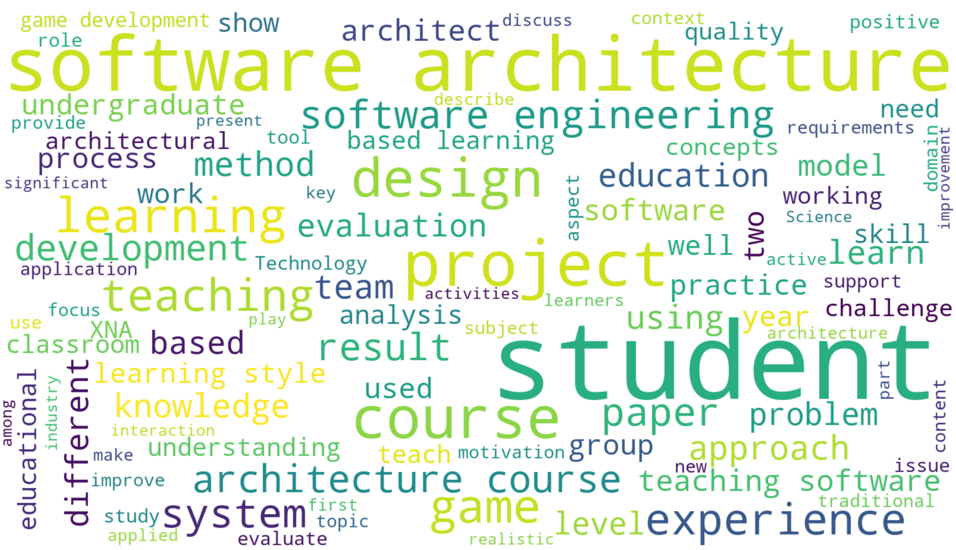


Figure 3.3: Word cloud generated from the abstracts.

3.3 Results and Discussion

It is observed that the earliest selected studies date back to 2005. Therefore, studies published between 2005 and 2025 were included in the analysis. Furthermore, the highest number of publications occurred in 2011, with five studies selected. These data are presented in Figure 3.4, which shows the annual distribution of studies included in the SMS.

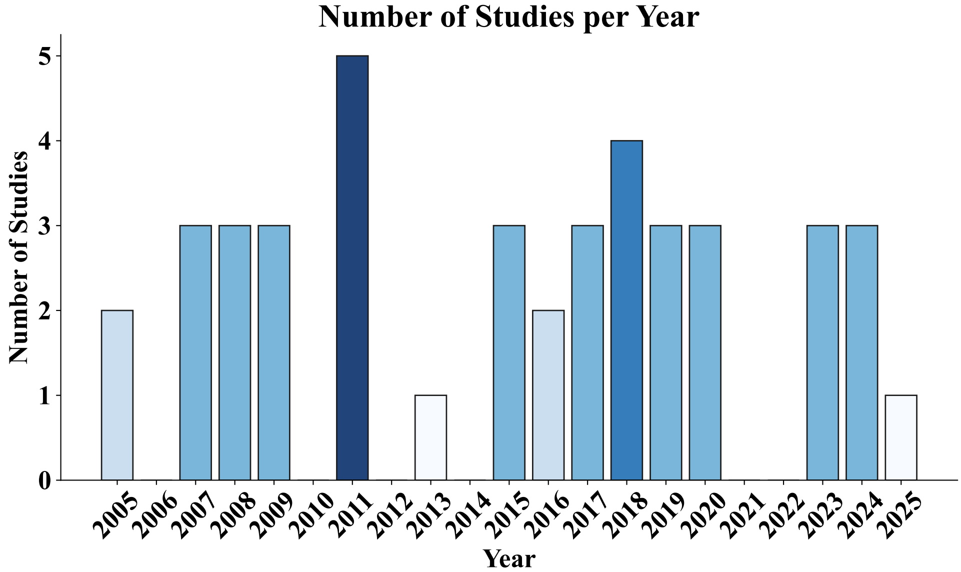


Figure 3.4: Number of published studies per year (2005–2025).

RQ1: Which topics have been taught in the field of Software Architecture?

Based on the analysis of 45 studies included in this SMS, it is possible to identify a wide variety of topics covered in SAE, revealing both the complexity of the field (GALSTER; ANGELOV, 2016) and the diversity of approaches adopted. The following provides a summary of the main themes identified in the publications.

One of the most recurrent topics is the teaching of software quality attributes (such as modifiability, usability, and testability) (URREGO; CORREAL, 2013; LELIS, 2024; CERVANTES et al., 2016), often addressed together with architectural evaluation techniques, particularly the ATAM (Architecture Tradeoff Analysis Method) (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017; VIDONI; MONTAGNA; VECCHIETTI, 2018; LAGO; VLIET, 2005). These elements are strongly associated with developing students' analytical and critical skills, particularly in making architectural decisions that involve trade-offs among conflicting quality attributes (BOER et al., 2019; LAGO et al., 2019; CERVANTES et al., 2016).

Another frequently cited topic concerns architectural and design patterns, such as MVC, Pipe-and-Filter, and Microservices (KEENAN; STEELE, 2011; GAST, 2008; CASTRO, 2023; WEI et al., 2020). The use of these patterns is typically associated with teaching architectural styles, component modeling, and the modular organization of complex systems (CASTRO, 2023; LAGO; VLIET, 2005). In several studies, the practical application of these patterns is demonstrated through games, software projects, or simulations, thereby reinforcing the applied and contextualized nature of the teaching (KEENAN; STEELE, 2011; WANG, 2011; LELIS, 2024; WANG; WU, 2011; BOER et al., 2019).

Architectural documentation, including the use of multiple views (4+1 model, C4, and those recommended by IEEE 1471), was also addressed (LAGO; VLIET, 2005; MANNISTO; SAVOLAINEN; MYLLARNIEMI, 2008; DEURSEN et al., 2017), with an emphasis on accurate stakeholder communication and the traceability of decisions throughout the system lifecycle (BOER; FARENHORST; VLIET, 2009; GAST, 2008). This topic is integrated with UML-based architectural modeling (class, component, and state diagrams) (RODRIGUES; WERNER, 2011; GRBAC; CAR; VUKOVIĆ, 2015) and with tools such as StarUML, ArchE (MCGREGOR et al., 2007), and Archinotes (URREGO; CORREAL, 2013).

The architectural design process encompasses defining architectural requirements, synthesizing and evaluating architectural solutions (GRBAC; CAR; VUKOVIĆ, 2015; WANG; STALHANE, 2005). In several studies, this process is taught using the Attribute-Driven Design (ADD) method (CERVANTES et al., 2016), which guides students in building architectures aligned with the system's quality goals.

Beyond technical content, some studies also emphasize social aspects, such as team collaboration (DEURSEN et al., 2017; VIDONI; MONTAGNA; VECCHIETTI, 2018), stakeholder communication (BOER; FARENHORST; VLIET, 2009; LAGO; VLIET, 2005), the role of the SA, and even soft skills such as leadership and negotiation (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017), essential elements for professional practice in real-world architectural contexts.

Finally, more specific topics are addressed depending on the course context or strategy, such as game architecture (KEENAN; STEELE, 2011; WANG, 2009; WANG; WU, 2011), embedded systems with robotics (KIM et al., 2008), distributed systems and microservices (WEI et al., 2020), DevOps and CI/CD (Continuous Integration and Continuous Deployment), and architectural design applied to specific domains such as big data, cloud computing, and IoT (GONÇALVES et al., 2020).

Taken together, these topics reveal the breadth of architectural concepts covered in education and the continued absence of emerging themes such as cloud-native systems, IoT, and DevOps (GALSTER; ANGELOV, 2016).

RQ2: What types of approaches have been adopted to teach Software Architecture?

The analysis of the studies included in this SMS reveals a diversity of approaches employed in SAE, with a particular emphasis on active and student-centered teaching strategies (DEURSEN et al., 2017; HIDALGO; ASTUDILLO; CASTRO, 2023). The predominant approach, as shown in Figure 3.5, is PBL, which is present in a large portion of the studies (VIDONI; MONTAGNA; VECCHIETTI, 2018; RUPAKHETI; CHENOWETH, 2015; AL-QORA'N; JAWARNEH; NGANJI, 2023), and requires the use of a broken axis to accommodate its higher frequency.

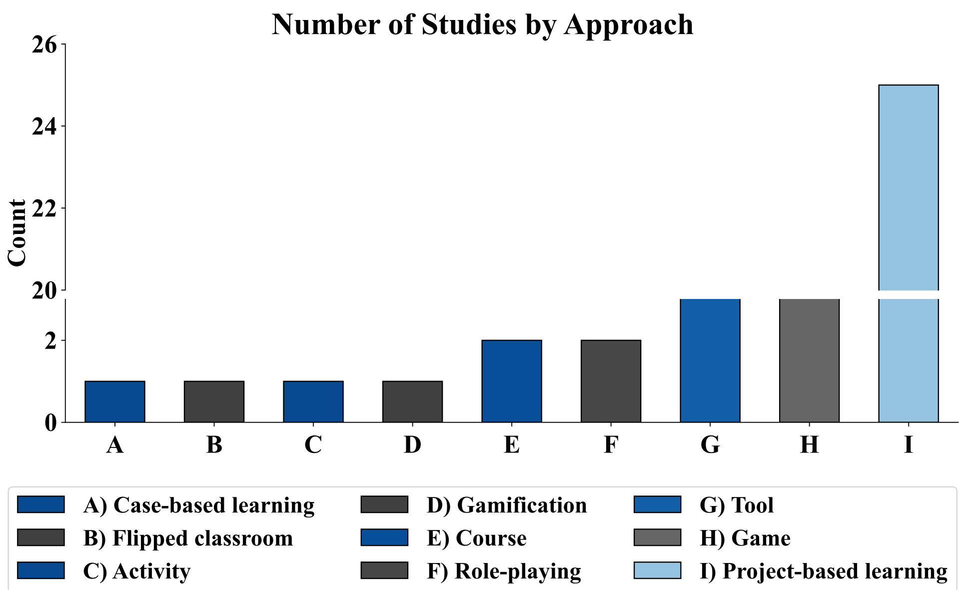


Figure 3.5: Distribution of approaches adopted in the studies.

PBL is applied broadly and in varied ways, from developing simulated systems (e.g., games, robots, and web platforms) (WANG, 2011; KIM et al., 2008; CAO; CAO, 2011) to redesigning or reverse-engineering existing systems (DEURSEN et al., 2017). Students, organized into teams, assume the roles of architect and stakeholder in realistic contexts (BOER; FARENHORST; VLIET, 2009; CHENOWETH; ARDIS; DUGAS, 2007). This approach is often combined with iterative design, review, refinement, and evaluation of architectural decisions (GAST, 2008; MANNISTO; SAVOLAINEN; MYLLARNIEMI, 2008).

In addition to PBL, other relevant strategies were identified in SAE. These include Case-Based Learning (CBL), which involves analyzing real-world scenarios or historical projects to understand architectural decisions and their consequences (OUH; IRAWAN, 2019; LIEH; IRAWAN, 2018b). Furthermore, the use of games and gamification elements, both digital and analog, stands out as a means to simulate the architectural decision-making process, apply methods such as ATAM, and foster reasoning about trade-offs (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017; CERVANTES et al., 2016; ARAÚJO et al., 2024). Examples include games such as DecidArch (BOER et al., 2019), D-LEARN (LIMA; MARQUES, 2024), and LEARN (LELIS, 2024).

Another adopted strategy is role-playing, where students assume distinct roles (architects, clients, and developers)in simulated projects (CHENOWETH; ARDIS; DUGAS, 2007; CASTRO, 2023). This approach contributes to both understanding the multiple perspectives involved in architectural decisions and developing interpersonal and technical communication skills (BOER; FARENHORST; VLIET, 2009; MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017).

The use of specialized software tools, such as ArchE (MCGREGOR et al., 2007), DECORA (CARVALHO et al., 2025), GARDIAN (KIM et al., 2008), and StarUML (KIM et al., 2008), was observed to support architectural design, evaluation, and documentation. These tools provide immediate feedback, guide students in applying quality methods such as ADD (CERVANTES et al., 2016) and COMET (KIM et al., 2008), and enable practical experimentation with architectural decisions in real time (URREGO; CORREAL, 2013; WU et al., 2009). Table 3.3 summarizes the main tools and games identified, highlighting their purposes and roles in Software Architecture Education.

Table 3.3: Tools and games used in SAE

Name	Type	Description	Reference
ArchE[6]	Tool	Architectural design assistant that suggests tactics, calculates the impact of decisions, and supports quality-based methods such as ADD.	(MCGREGOR et al., 2007)
DECORA[7]	Tool	Decision support system for architectural decisions, with an interactive interface and recommendations based on questionnaires.	(CARVALHO et al., 2025)
GARDIAN	Tool	Verifies the compliance of architectural models with the COMET methodology, used in conjunction with tools like StarUML.	(KIM et al., 2008)
StarUML[8]	Tool	UML modeling tool used for building and documenting software architectures.	(KIM et al., 2008)
Archinotes[9]	Tool	Collaborative platform for architectural documentation and communication, integrating requirements, viewpoints, and prioritizations.	(URREGO; CORREAL, 2013)
XQUEST	Tool	Framework that facilitates the development of architecture-focused games, integrating patterns and reusable components.	(WU et al., 2009)
Sheep	Tool	Android extension serving as a secondary stimulus in teaching via dual-stimulation theory, providing technical support for the project.	(WU et al., 2010)
LEARN[10]	Board game	Educational card game using true/false questions that promotes concept review and collaborative learning.	(LELIS, 2024)
D-LEARN[11]	Digital game	Digital games combining quizzes with board and card dynamics, focusing on architectural decisions and gamification.	(LIMA; MARQUES, 2024)
DecidArch[12]	Card game	Physical card game simulating architectural decisions based on the ADD method, including role-playing elements.	(BOER et al., 2019)
ATAM-RPG	RPG	Role-playing game simulating an ATAM meeting, promoting technical learning and social skills development.	(MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017)
Smart Decisions[13]	Board game	Simulates the architectural design process using structured rounds, decision cards, and group discussion.	(CERVANTES et al., 2016)
Kahoot[14]	Tool / Gamif.	Interactive quiz platform that reinforces theoretical concepts in a playful and competitive manner.	(ARAÚJO et al., 2024; LIMA; MARQUES, 2024)

Additionally, some approaches are structured as courses with a practical focus, while still retaining traditional characteristics (LAGO; VLIET, 2005). In these initiatives, theoretical content is combined with activities such as workshops, controlled exper-

[6] www.sei.cmu.edu/architecture/arche.html
[7] https://decora-front.vercel.app/
[8] https://staruml.io/
[9] https://archistudentnotes.com/
[10] https://zenodo.org/records/11200966
[11] https://d-learn.vercel.app/game
[12] https://github.com/S2-group/DecidArch
[13] https://smartdecisionsgame.com/
[14] https://kahoot.it/

iments (WANG; ARISHOLM; JACCHERI, 2007), lectures with industry experts (MANNISTO; SAVOLAINEN; MYLLARNIEMI, 2008), and Post-Mortem Analysis sessions (PMA) (WANG, 2009; WANG; STALHANE, 2005), providing a more applied learning environment that closely reflects real market conditions (WEI et al., 2020).

The flipped classroom approach was also identified in some studies, reorganizing instructional time (GONÇALVES et al., 2020). In this strategy, students access theoretical content beforehand through videos, readings, or online activities, and use class time for discussions, problem-solving, and hands-on activities guided by the instructor. Results indicate that this model contributes to student engagement, stimulates critical thinking, and supports personalized learning (GONÇALVES et al., 2020).

Finally, many analyzed proposals adopt blended approaches, combining different strategies (ANGELOV; BEER, 2015; ALPEROWITZ et al., 2017). The integration of PBL with gamification, role-playing, and software tools, or combining theory and practice through labs and projects, demonstrates an effort to provide broader education, tailored to the multiple demands of the teaching-learning process in Software Architecture (LELIS, 2024; CERVANTES et al., 2016).

These approaches are guided by experiential learning principles, emphasizing critical reflection, problem-solving, and practical experimentation (LIEH; IRAWAN, 2018a). Overall, approaches to SAE have evolved toward active, student-centered, experiential learning that aligns teaching with professional practice and market demands (LIEH; IRAWAN, 2018a; LAGO; VLIET, 2005).

RQ3: Is there evidence of the effectiveness of these approaches in the student teaching-learning process?

The empirical evidence extracted from the analyzed studies indicates that the approaches adopted for SAE have contributed significantly to the student teaching-learning process, both in terms of knowledge acquisition and the development of technical, interpersonal, and cognitive skills (CARVALHO et al., 2025; WEDEMANN, 2018; CERVANTES et al., 2016; DEURSEN et al., 2017).

Various assessment strategies were used to verify effectiveness, including Likert-scale questionnaires (OUH; IRAWAN, 2019; WU et al., 2009), feedback forms (IMI, IMMS and MEEGA+), student self-assessments (WANG, 2011), academic performance analysis (exams, projects, grades) (WANG, 2011; WEDEMANN, 2018), code reviews (DEURSEN et al., 2017), and PMA (WANG, 2009; WANG; STALHANE, 2005). These instruments provided qualitative and quantitative evidence supporting the effectiveness of the active and innovative approaches implemented (CARVALHO et al., 2025; OUH; GAN; IRAWAN, 2020; WANG; ARISHOLM; JACCHERI, 2007).

Students reported a high impact on learning and increased confidence in justifying and documenting architectural decisions, particularly after engaging in role-playing activities and interacting with stakeholders (BOER; FARENHORST; VLIET, 2009; CHENOWETH; ARDIS; DUGAS, 2007; MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017). Software tools such as ArchE (MCGREGOR et al., 2007), DECORA (CARVALHO et al., 2025), Sheep (WU et al., 2010), and XQUEST (WU et al., 2009) have helped reduce the cognitive load associated with modeling and programming, enabling a greater focus on architectural concepts. Qualitative evaluations highlighted the role of these tools in fostering deeper reflection and student autonomy (URREGO; CORREAL, 2013; WU et al., 2009).

The use of simulations and games such as DecidArch (BOER et al., 2019; LAGO et al., 2019), D-LEARN (LIMA; MARQUES, 2024), and ATAM-RPG (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017) proved effective in stimulating student interest, promoting meaningful learning, critical thinking, and engagement (CERVANTES et al., 2016; ARAÚJO et al., 2024; LELIS, 2024). MEEGA+ evaluations model indicated gains in motivation, challenge, and social interaction, as well as high satisfaction with the content and its applicability (LIMA; MARQUES, 2024; LELIS, 2024).

The flipped classroom yielded statistically significant gains in content acquisition and understanding of essential concepts, improving classroom interaction and the quality of student deliverables (GONÇALVES et al., 2020). PBL provided realistic context, collaboration, and practical concept application, receiving positive evaluations from students in terms of engagement and performance (VIDONI; MONTAGNA; VECCHIETTI, 2018; AL-QORA'N; JAWARNEH; NGANJI, 2023; RUPAKHETI; CHENOWETH, 2015; DEURSEN et al., 2017).

PMA sessions were reported as productive by both students and instructors, providing more useful feedback than traditional assessments and preparing students for the professional environment (WANG, 2009; WANG; STALHANE, 2005). Additionally, practices such as architectural documentation, decision traceability, and trade-off analysis were valued as effective ways to consolidate learning (GAST, 2008; LAGO; VLIET, 2005; MANNISTO; SAVOLAINEN; MYLLARNIEMI, 2008).

Although some studies did not find significant differences in final grades between active and traditional approaches (WANG, 2011; WANG; WU, 2011), qualitative reports indicate gains in motivation, autonomy, perceived learning, engagement, and professional preparedness (CARVALHO et al., 2025; WEDEMANN, 2018; ARAÚJO et al., 2024; GALSTER; ANGELOV, 2016). In many cases, students with little prior experience were able to keep up with or surpass more experienced peers, suggesting that these methodologies compensated for experience gaps (OUH; IRAWAN, 2019; LIEH; IRAWAN, 2018b).

In summary, the innovative approaches analyzed show clear evidence of effectiveness in SAE. They enhance comprehension and promote changes in how students think and act in the learning process (LIEH; IRAWAN, 2018a; HIDALGO; ASTUDILLO; CASTRO, 2023). Although results vary with institutional context, student profile, and tool maturity, the data from this SMS suggest that these approaches offer promising pathways for more effective, motivating, and industry-aligned teaching (CARVALHO et al., 2025; WU et al., 2009; OUH; GAN; IRAWAN, 2020).

Overall, the evidence suggests that innovative approaches not only enhance technical understanding but also promote motivation, autonomy, and professional readiness (LIMA; MARQUES, 2024; ARAÚJO et al., 2024; WEDEMANN, 2018).

RQ4: What are the main challenges in Software Architecture Education?

SAE faces challenges, many of which stem from the complexity of the content and student profiles (HIDALGO; ASTUDILLO; CASTRO, 2023). A key obstacle is students' difficulty in developing architectural views and relating stakeholder concerns, which requires handling problems with no single solution (LAGO; VLIET, 2005; BOER; FARENHORST;

VLIET, 2009). This demands a mindset that goes beyond technical knowledge, involving social, communicative, and analytical skills such as critical thinking, negotiation, and consensus building (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017; VIDONI; MONTAGNA; VECCHIETTI, 2018).

The abstract and highly conceptual nature of architecture makes initial comprehension difficult, especially for students without prior practical experience or technical leadership (OUH; IRAWAN, 2019; LIEH; IRAWAN, 2018a; CARVALHO et al., 2025). Many struggle to connect theory with real-world situations and to understand the social and mediating role of the architect between clients and technical teams (BOER; FARENHORST; VLIET, 2009; HIDALGO; ASTUDILLO; CASTRO, 2023; ALPEROWITZ et al., 2017). The lack of contextualized practical experiences and teaching materials widens this gap, demanding methods that effectively integrate theory and practice (GALSTER; ANGELOV, 2016; RUPAKHETI; CHENOWETH, 2015; CARVALHO et al., 2025).

Another challenge lies in time management and workload in courses, which are often short and involve extensive and complex activities, generating overload and demotivation (BOER; FARENHORST; VLIET, 2009; OUH; GAN; IRAWAN, 2020). This includes providing sufficient guidance for the architectural design process without restricting solution diversity, while maintaining rigor in documenting decisions and utilizing tools that are not always intuitive or adapted to the academic context (URREGO; CORREAL, 2013; MCGREGOR et al., 2007; MANNISTO; SAVOLAINEN; MYLLARNIEMI, 2008).

Coordination and communication among groups present difficulties, particularly in collaborative projects where alignment between architects and stakeholders is essential (DEURSEN et al., 2017; CHENOWETH; ARDIS; DUGAS, 2007; BOER; FARENHORST; VLIET, 2009). Evaluation tools and methods, such as ATAM, are difficult to apply in academic settings due to a lack of adaptation and students' inexperience with these processes (MONTENEGRO; ASTUDILLO; ÁLVAREZ, 2017; VIDONI; MONTAGNA; VECCHIETTI, 2018). Another critical aspect is students' difficulty in dealing with multiple levels of abstraction, traceability, trade-offs between quality attributes, and the need to explicitly express design without relying on implementation (GAST, 2008; MANNISTO; SAVOLAINEN; MYLLARNIEMI, 2008; CARVALHO et al., 2025; UR-

REGO; CORREAL, 2013). Concepts such as extensibility, concurrency, and parallelism are complex and require more experience, which is not always feasible in traditional courses (CAO; CAO, 2011; WEI et al., 2020).

Finally, there are challenges related to adapting teaching to the diverse characteristics of students and the academic context, such as heterogeneous profiles, limited technological resources, and the need to reconcile scientific and educational objectives without diminishing student motivation (HIDALGO; ASTUDILLO; CASTRO, 2023; CARVALHO et al., 2025; WANG; ARISHOLM; JACCHERI, 2007). This implies the need for models that respect individual learning paces, promote active participation, and integrate technical and interpersonal skills to prepare students for the real-world challenges of SA (LIEH; IRAWAN, 2018a; HIDALGO; ASTUDILLO; CASTRO, 2023; ARAÚJO et al., 2024; GONÇALVES et al., 2020).

3.4 Challenges and Future Directions

Our analysis reveals that, despite notable progress, several persistent challenges limit the effectiveness, scalability, and industry alignment of SAE. The following challenges are grounded in the evidence obtained in our SMS (see RQ1–RQ4) and identify concrete avenues for future research and curricular action, as described below and summarized in Figure 3.6:

1. **Bridging abstraction and practice.** Students frequently struggle to reason across multiple architectural views and reconcile stakeholder concerns (RQ4). This difficulty is compounded by the predominance of short-term, course-limited interventions (RQ3), which provide insufficient exposure to authentic design contexts. In future work, the scientific community should investigate cognitive scaffolding techniques (e.g., stepwise concretization, worked examples) and practice-oriented formats (e.g., longitudinal design studios, industrially partnered capstone projects, tool-supported modeling exercises) that make abstraction tangible while preserving analytical rigor;

2. **Longitudinal evaluation of impact.** The current evidence base relies on per-

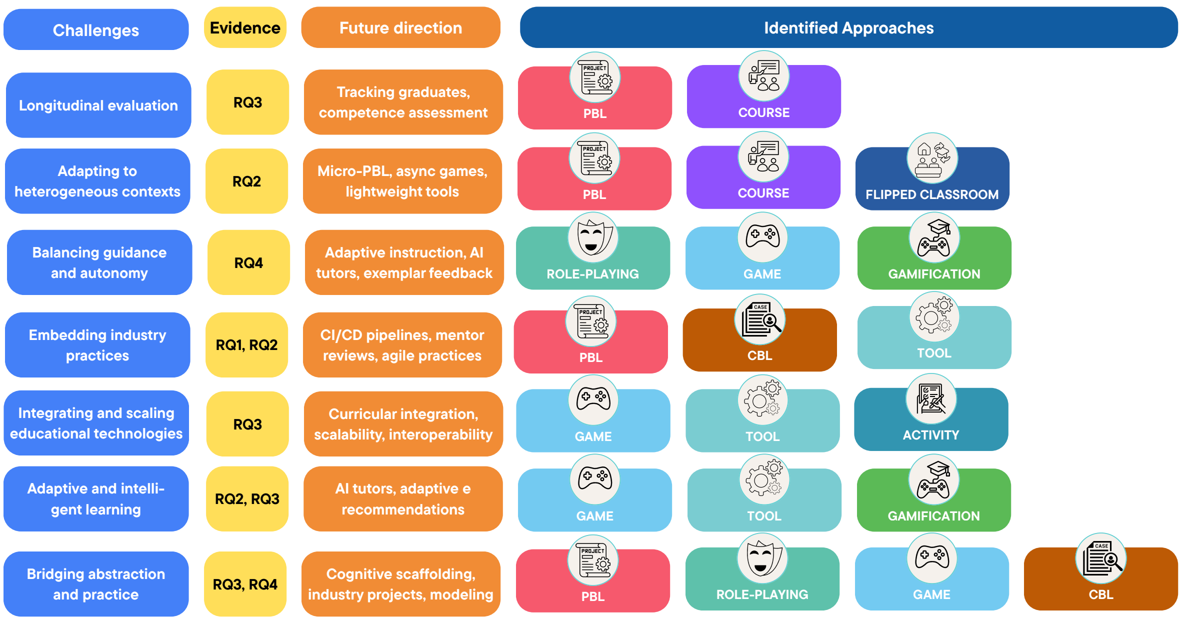


Figure 3.6: Challenges in SAE with related evidence, future directions, and approaches.

ception surveys and single-course assessments (RQ3), limiting our understanding of long-term effects on professional competence. Research should develop longitudinal evaluation models and instruments that track graduates over time, correlating educational treatments with sustained skill acquisition, employability metrics, and on-the-job performance;

3. **Adapting interventions to heterogeneous contexts.** Institutional constraints (limited contact hours, heavy workloads, uneven access to technology) hinder the adoption of active methods such as PBL and games (RQ2). There is a need for adaptive, low-cost, and scalable implementation (micro-PBL modules, asynchronous game-based activities, and lightweight tooling) that can be tailored to institutions with diversified resources;

4. **Balancing guidance and autonomy.** Studies have identified a persistent tension between the provision of adequate scaffolding and the promotion of student creativity (RQ4). Excessive instructional guidance may constrain opportunities for design exploration, whereas insufficient support can lead to cognitive overload among novices. Future work should examine adaptive instructional approaches and decision-support mechanisms that can dynamically calibrate the degree of guidance

in relation to learner profiles and developmental trajectories. Promising avenues include the integration of AI-assisted tutoring systems and exemplar-based feedback strategies;

5. **Embedding authentic industry practices.** Although teamwork and role simulations are common, curricula often lag in representing contemporary practices (e.g., DevOps, cloud-native patterns, microservices) (RQ1, RQ2). Research should evaluate the classroom integrations of authentic industry workflows (for example, continuous delivery pipelines and architecture reviews with industry mentors) and measure their impact on readiness for professional practice.

6. **Systematic integration and scaling of educational technologies.** Tools and games (e.g., ArchE, DECORA, DecidArch, ATAM-RPG) demonstrate value (RQ3), but remain fragmented across curricula. Future work should investigate strategies for sustainable curricular integration, interoperability, and deployment at scale (including in distributed/remote learning environments), and produce design patterns for tool adoption that align with course learning objectives;

7. **Leveraging adaptive and intelligent learning.** Emerging AI capabilities offer personalized learning (adaptive recommendations, automated feedback, intelligent tutoring). Research at the intersection of SAE and learning technologies should involve the development and evaluation of AI-driven interventions that support distributed collaboration, reduce instructor workload, and tailor instruction to individual needs.

In short, practitioners and educators should: (i) prioritize longitudinal, practice-oriented learning experiences in the design of SA curricula; (ii) implement modular and scalable approaches that accommodate institutional constraints; and (iii) employ evaluation instruments capable of capturing long-term, transferable learning outcomes. We believe that addressing the challenges above will increase the likelihood that graduates acquire a deeper understanding of SA and are better prepared for the demands of contemporary industry.

Addressing these challenges is not optional: it is essential to ensure that future curricula remain effective, scalable, and aligned with the rapidly evolving demands of software engineering practice.

3.5 Threats to Validity of the SMS

Despite efforts to ensure methodological rigor and reliability of the results of this SMS, it is necessary to acknowledge that certain threats to validity and limitations may have influenced the findings and interpretations. Identifying and discussing these aspects is essential to guide appropriate interpretation of results and delimit the scope of conclusions (MOURÃO et al., 2020). The main threats to validity identified in this work, along with the strategies adopted to mitigate them, are discussed in the following subsections.

A relevant threat to **internal validity** is associated with human factors, which may have influenced the process of data extraction and interpretation from the primary studies. In some cases, the information reported by the authors was not sufficiently detailed, requiring interpretations by the reviewers. To mitigate this risk, all stages of study selection and data extraction were conducted in pairs, ensuring independent review and cross-checking of the results. Discrepancies were discussed in consensus meetings, reducing subjectivity and increasing the reliability of the process. Additionally, the final data extraction was carefully reviewed to ensure consistency.

External validity may have been affected by the choice of databases used, although these were selected for their representativeness and relevance to the investigated topic. Even though four widely recognized databases were consulted (IEEE Xplore, Scopus, Engineering Village, and SOL), it is possible that relevant studies published in other repositories were not retrieved. This limitation may restrict the generalizability of the obtained results. However, efforts were made to minimize this risk by selecting databases with broad coverage of Software Engineering and Computing, increasing the likelihood of capturing significant studies for the SMS scope.

One of the main threats to **construct validity** refers to the possibility of missing relevant studies due to the search string. Some works might not have been retrieved if they did not contain the specified terms in their titles, abstracts, or keywords. To mitigate this threat, the search string was constructed through a systematic process that involved identifying keywords present in the literature, validating them against previously known central studies, and refining them through several iterations in the Scopus database. This procedure enabled adjustments and refinements to the string, ensuring greater coverage and precision in study retrieval. Furthermore, inclusion and exclusion criteria were clearly defined and applied by two reviewers in parallel, reinforcing consistency in article selection.

Conclusion validity may have been compromised in situations where the primary studies presented missing, incomplete, or unclear information, requiring additional interpretation by the reviewers. This factor represents a risk to the reliability of the conclusions drawn. To mitigate this issue, all stages of the SMS, including the formulation of RQ, definition of the search string, study selection, and data extraction, were carefully documented, collectively discussed, and validated by experienced reviewers in the field. Additionally, the analysis process was done collaboratively, with paired reviews and consensus-based decisions to reduce individual biases. Although systematic efforts were made to mitigate these risks, it is acknowledged that they cannot be fully eliminated.

3.6 Conclusion of the SMS

This SMS synthesized key topics, approaches, and challenges in SAE literature. For RQ1, the analysis showed diverse teaching content, from architectural concepts, quality attributes, and trade-offs to methodologies such as ATAM, modeling, documentation, and stakeholder communication, highlighting SAE complexity and the need for curricula integrating technical and communicative dimensions.

The results of RQ2 indicate a predominance of active approaches, particularly PBL, often complemented by educational games, simulations, case studies, and collaborative activities. While traditional lecture-based methods remain present, the trend is toward hybrid and experiential models that align with students' professional preparation.

Concerning RQ3, the evidence shows that innovative approaches, such as games, role-playing, computational tools, and flipped classrooms, effectively enhance student learning by increasing motivation, engagement, critical thinking, autonomy, and technical and interpersonal competencies while bridging theory and professional practice.

Finally, in relation to RQ4, this study highlights persistent challenges: the difficulty of mastering abstract concepts and complex problems, the scarcity of authentic practice opportunities, and the time constraints and workload limitations of courses. These findings highlight the importance of tailoring tools and strategies to academic contexts while balancing guidance and student autonomy.

Taken together, these insights underscore the need for innovative models that integrate theory and practice, promote critical reflection, and develop the technical, analytical, and interpersonal skills essential to the professional role of software architects. They also emphasize the importance of creating accessible resources and designing robust assessment strategies that can capture both immediate and long-term learning outcomes.

As a contribution, this SMS updates and extends previous secondary studies and consolidates evidence to inform curriculum design, instructional practices, and educational policy in SA. For future work, we identify opportunities to: (i) investigate the impact of active methodologies on professional competence; (ii) explore scalable uses of tools, games, and hybrid models; and (iii) strengthen alignment between academic practices and industry demands, particularly regarding agile methods and emerging architectures.

By addressing these directions, the research community can advance toward more effective, sustainable, and contextually relevant approaches to SAE, ultimately preparing students to meet the evolving challenges of the field. We believe that advancing along these directions is key to shaping a generation of software architects capable of meeting the demands of an increasingly complex, dynamic, and industry-driven landscape.

3.7 Final Considerations

The findings on the effectiveness of active learning strategies (RQ2 and RQ3), together with the persistent challenges related to abstraction, student participation, and integration of technical and social skills (RQ4), served as the foundation and directly informed the conception of the proposed game. Therefore, as a direct consequence of the diagnosis provided by this SMS, the following chapter is dedicated to presenting the Architectural Stories Game, detailing its development, mechanics, and how its pedagogical structure was engineered to solve the architectural education problems documented in this study.

4 Architectural Stories Game

Architectural Stories is a non-digital investigative deduction game designed to support SAE by adapting the narrative-deduction mechanics of *Black Stories* to the context of software design. The game operationalizes the concept of Architectural situations to help students transition from abstract theory to a practical, analytical skill set focused on diagnosing root causes in realistic system failure scenarios, and its design is grounded in our previously submitted work (MENEZES; VALLE; OLIVEIRA, 2026a).

Furthermore, the conception of the game is directly informed by the evidence synthesized in the SMS presented in Chapter 3, specifically addressing the need for active learning strategies, increased student engagement, and the integration of technical and analytical skills, which are identified as persistent challenges in SAE.

To detail the development and structure of this educational artifact, this chapter is organized as follows: Section 4.1 defines the educational objectives and targeted competencies; Section 4.2 describes the physical game components and the design of the Mystery Cards; Section 4.3 presents the scoring system and progress tracking; Section 4.4 outlines the supplementary instructional materials; and Section 4.5 details the gameplay structure and role dynamics;

4.1 Objectives and Targeted Competencies

The design of Architectural Stories is directly grounded in the learning objectives defined in Phase 2 of the methodology on Section 1.4. These objectives aim to move students from a passive, conceptual understanding of SA toward an active, diagnostic, and argumentative application of architectural knowledge.

The first learning objective is to enhance pattern recognition. In the game, this objective is operationalized through open-ended narrative enigmas that describe software system failures without explicitly naming architectural problems. Players must interpret these narratives and identify architectural symptoms, such as low cohesion, excessive coupling, or quality attribute violations, to associate contextual clues with appropriate architectural patterns.

The second objective is to foster deductive reasoning. Throughout the gameplay, students are required to formulate hypotheses and iteratively refine them based on evidence revealed through questioning and subsequent analysis. This process mirrors diagnostic reasoning, as players must logically link an identified problem to a specific architectural solution, mapping symptoms to patterns through structured deduction rather than trial and error.

Finally, the game encourages theoretical argumentation and technical communication. To propose a complete solution, players must articulate their architectural choices using precise technical vocabulary and justify their decisions by explicitly discussing design trade-offs. This requirement reinforces the use of formal architectural concepts and supports the development of communication skills essential for explaining and defending architectural decisions in professional contexts.

4.2 Game Components and Physical Artifacts

Architectural Stories is designed for play in groups of three to five participants, a configuration that supports collaboration while maintaining individual accountability. The game consists of the following physical components: a central score-tracking board, 20 Mystery Cards, colored pawns, an instruction manual, and a set of pattern reference cards.

The decision to adopt a non-digital format was intentional. By avoiding technological mediation, the game emphasizes face-to-face discussion, collective reasoning, and social interaction, which are essential aspects of collaborative architectural work. Additionally, the physical format facilitates adoption in classroom environments without requiring specific technical infrastructure.

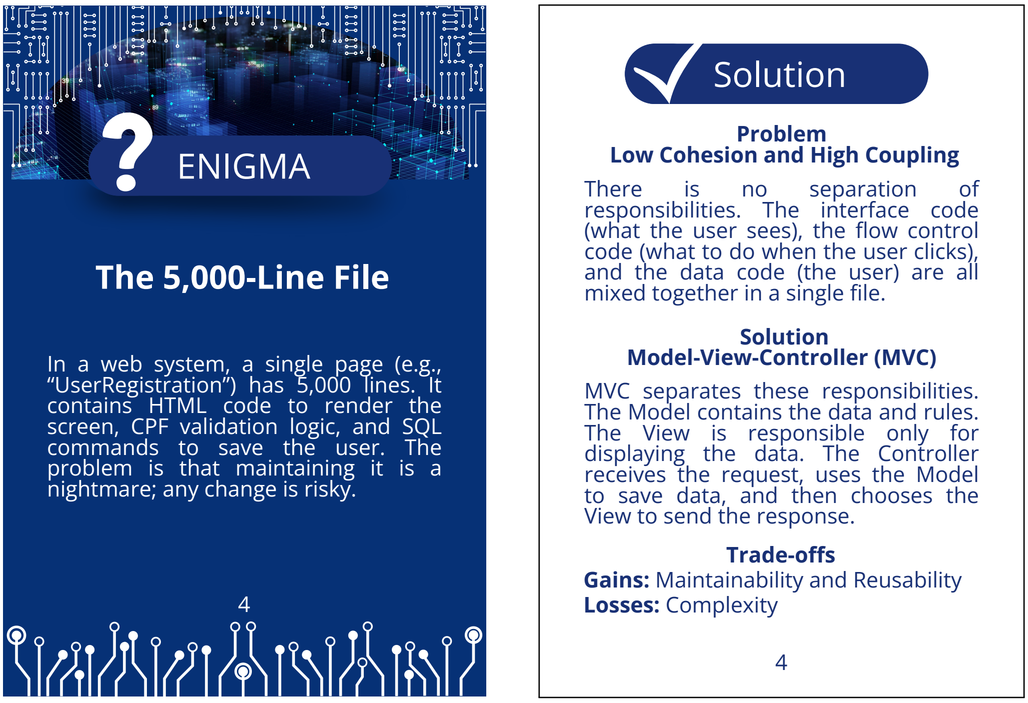
4.2.1 Mystery Cards

The Mystery Card is the core pedagogical artifact of the game. The complete collection of the 20 cards developed for this study is available in Appendix A. Each card is composed of two complementary sides: the Enigma and the Solution.

The Enigma, visible to all players, presents a short narrative describing a failure, anomaly, or problematic behavior observed in a software system. These narratives are intentionally concise yet rich in information, providing sufficient context to stimulate inquiry while preserving ambiguity. Realistic development practices and common industry problems inspire the scenarios.

Figure 4.1(a) illustrates an example enigma entitled “The 5,000-Line File,” which describes a monolithic web page that combines presentation logic, validation, and database access within a single file. The narrative implicitly exposes symptoms such as low cohesion, high coupling, and poor maintainability, without explicitly naming these issues.

The Solution, visible only to the Architecture Master (AM), contains the authoritative resolution of the enigma. It explicitly identifies the architectural root cause and prescribes an appropriate architectural pattern or principle. As shown in Figure 4.1(b), the solution to “The 5,000-Line File” diagnoses the problem as a violation of separation of concerns and recommends the application of the MVC pattern, outlining its benefits and trade-offs. This asymmetric information structure is a fundamental game mechanic, as it drives inquiry, discussion, and collaborative reasoning among players.



(a) The Enigma. (b) The Solution.

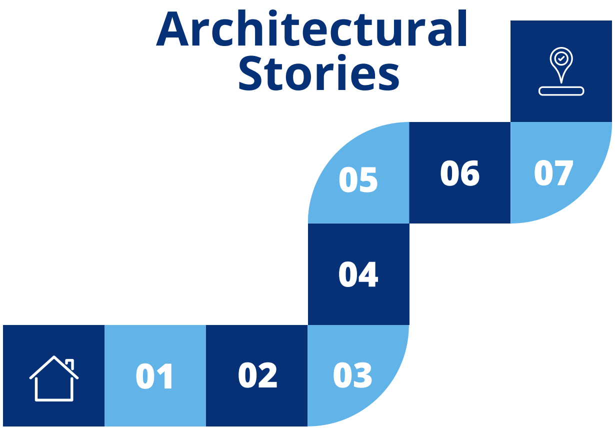
Figure 4.1: Enigma and solution used in the activity.

4.3 Scoring System and Progress Tracking

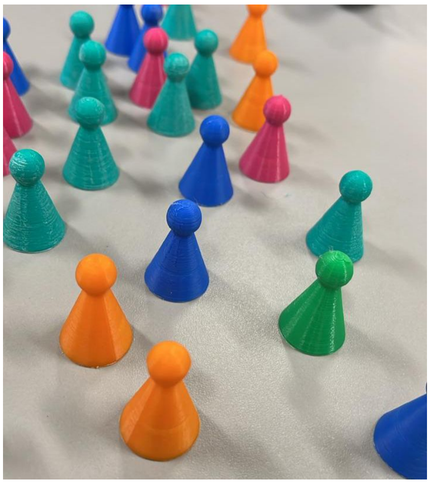
To support engagement and provide continuous feedback, the game incorporates the Insight Points Board, illustrated in Figure 4.2(a). This board centrally tracks player performance and serves as a shared visual reference throughout the session.

Progress on the board is represented by colored, 3D-printed pawns (Figure 4.2(b)), with each pawn corresponding to a single participant. This design choice reinforces individual accountability while preserving group interaction.

The scoring system is intentionally simple. Correct solutions award one Insight Point, while incorrect accusations result in a penalty of one point. This mechanism encourages careful reasoning, discourages random guessing, and reinforces the importance of evidence-based decision-making.



(a) Insight Points Board.



(b) Colored pawns.

Figure 4.2: Insight Points Board and Colored pawns.

4.4 Supplementary Instructional Materials

To ensure consistency and pedagogical support, Architectural Stories includes supplementary instructional materials.

The Instruction Manual (provided in Appendix B) provides detailed guidance on game setup, rules, roles, phases, and scoring. Its clarity was a design priority, particularly because one of the evaluation objectives was to assess whether students could understand

and play the game autonomously without instructor intervention.

The Pattern Reference Cards serve as theoretical scaffolding throughout gameplay, providing a comprehensive description of the underlying architectural patterns and principles, as presented in Annex A. The cards summarize a specific architectural pattern, including its intent, structural characteristics, typical applications, and trade-offs. During investigations, players consult these cards to validate hypotheses and ground their reasoning in formal architectural knowledge.

This integration ensures that the game emphasizes analytical reasoning rather than rote memorization, allowing students to verify their conclusions against established architectural theory. All materials are publicly available to support replication and reuse[15].

4.5 Gameplay Structure and Role Dynamics

Gameplay follows a rotating-role structure involving one AM and multiple investigators. This design realizes the hybrid competitive and cooperative dynamics in the methodology.

As illustrated in Figure 4.3, each round begins with a preparation phase, during which the AM selects a Mystery Card and reviews the solution. The AM then presents the Enigma to the group and moderates the session by answering closed-ended questions with "Yes," "No," or "Irrelevant."

Investigators collaboratively analyze these responses to formulate and refine hypotheses. At any point, a player may attempt to present a Complete Solution, which must clearly articulate both the architectural diagnosis and the corresponding pattern. Each round concludes with a debriefing phase led by the AM, during which the narrative is explicitly connected to the architectural concepts involved. This reflective moment reinforces learning by making implicit reasoning explicit and addressing misconceptions. The game ends when a player reaches the end of the Insight Points Board or when the predefined time limit is reached.

[15] https://tinyurl.com/55jr3vhz

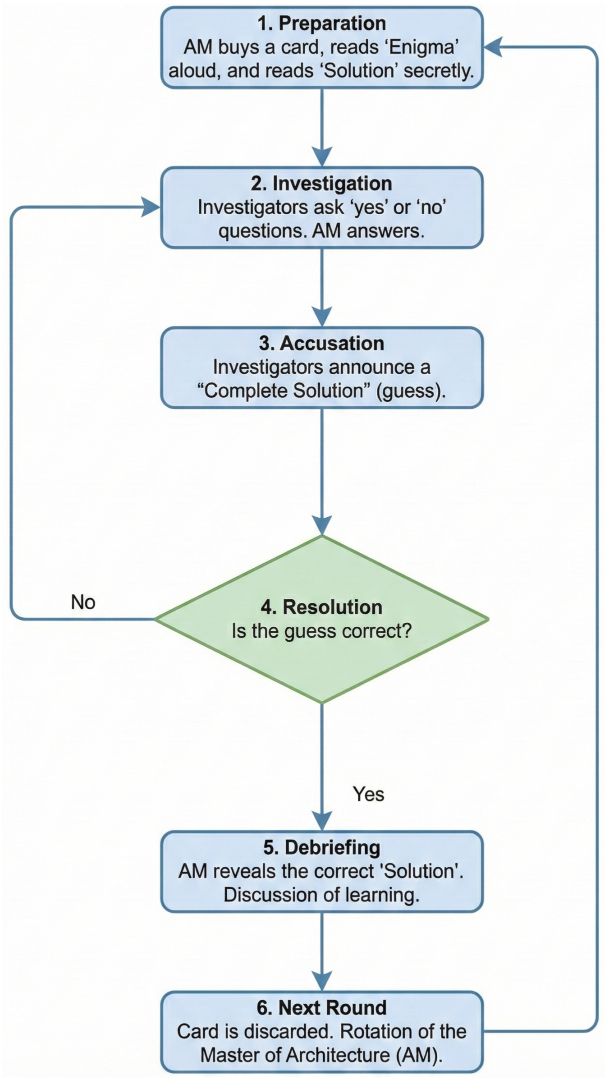


Figure 4.3: Game round flowchart, from preparation to debriefing.

4.6 Final Considerations

Through its narrative-driven structure, collaborative mechanics, and emphasis on architectural reasoning, Architectural Stories represents a research-driven educational artifact grounded in the evidence synthesized by the Systematic Mapping Study. By translating recurrent challenges in SAE into an engaging and structured gameplay experience, the game aims to support deeper understanding, sustained engagement, and the development of competencies essential to professional architectural practice. However, to confirm that this design effectively meets its pedagogical goals, its impact must be empirically assessed. Therefore, the following chapter describes the experimental setup, the data collection methods, and the results obtained from the evaluation.

5 Evaluation of Architectural Stories

Following the development of the Architectural Stories Game, it was essential to empirically validate its effectiveness as a pedagogical tool within a real educational setting. This chapter presents the evaluation phase of the research. The primary goal is to assess whether the game successfully addresses the challenges of abstraction and engagement in SAE by measuring player experience, intrinsic motivation, and instructional appeal. Through a multi-instrument approach involving the MEEGA+, IMI, and IMMS, this evaluation offers a multifaceted perspective on how the game impacts the learning of architectural patterns and students' motivation to engage with complex software engineering concepts.

To detail this evaluation process, the remainder of this chapter is organized as follows: Section 5.1 outlines the evaluation planning, including the selection of instruments and study design; Section 5.2 describes the execution of the gameplay sessions and data collection procedures; Section 5.3 presents the results and data analysis, providing a deep dive into participant profiles and instrument-specific findings; and Section 5.4 addresses the threats to the validity of the study.

5.1 Evaluation Planning

This evaluation is characterized as an evaluative case study with a descriptive and exploratory nature. The primary objective is to assess students' perceptions regarding the game's quality, usability, and player experience, as well as their intrinsic motivation and the motivational effectiveness of the instructional material.

To achieve this, we employed a multi-instrument approach. First, we employed MEEGA+ as the primary instrument for game quality assessment. MEEGA+ was chosen because it is a systematically developed and statistically validated model specifically tailored for computing education games. Its comprehensive structure, which decomposes quality into Usability and Player Experience factors, provides sufficient breadth to evaluate the game. Furthermore, the instrument's high reliability (Cronbach's alpha $\alpha = 0.928$) ensures rigorous data collection regarding student perceptions. While the standard MEEGA+ model contains 35 items, we applied the specific adaptation for non-digital games prescribed by the authors. Items 10, 11, and 12, which evaluate software customization and error protection, were excluded as they are irrelevant to card games. Consequently, the final MEEGA+ component consisted of 32 quantitative items using a 5-point Likert scale, complemented by open-ended questions to collect qualitative feedback.

To provide a deeper analysis of the students' subjective experience, we also utilized the IMI. The IMI is a multidimensional measurement device intended to assess participants' subjective experience related to a target activity. It evaluates specific subscales, including Interest/Enjoyment, Perceived Competence, Effort/Importance, Pressure/Tension, Perceived Choice, and Value/Usefulness, while the individual performs the activity.

Finally, to evaluate the motivational aspects of the game specifically as an instructional material, we applied the IMMS. The IMMS is based on the ARCS motivational design model, assessing four specific domains: Attention, Relevance, Confidence, and Satisfaction. This instrument consists of 36 statements where participants indicate the truthfulness of each statement in relation to the material studied, using a 5-point Likert scale ranging from "Not true" to "Very true"

A single gameplay session was conducted to fit within the constraints of the standard course curriculum. The duration of the gameplay session (approximately 30 minutes) was considered sufficient for participants to experience multiple rounds of the game. After the gameplay, an additional 40 minutes were allocated for students to complete the evaluation forms, which consolidated the MEEGA+, IMI, and IMMS items into a single data collection process.

5.2 Evaluation Execution

The evaluation was conducted during a scheduled Software Engineering class with 34 undergraduate students. The flow of the evaluation process is illustrated in Figure 5.1. Regarding prior knowledge, students had been introduced to fundamental architectural concepts in previous course modules; however, their familiarity with specific patterns varied, and no specific training on the game mechanics was provided beforehand. Participants organized themselves into independent groups of three to five members and received the game materials and rule manual. To assess the learnability of the materials, groups were instructed to learn the gameplay mechanics autonomously, relying exclusively on the provided manual.

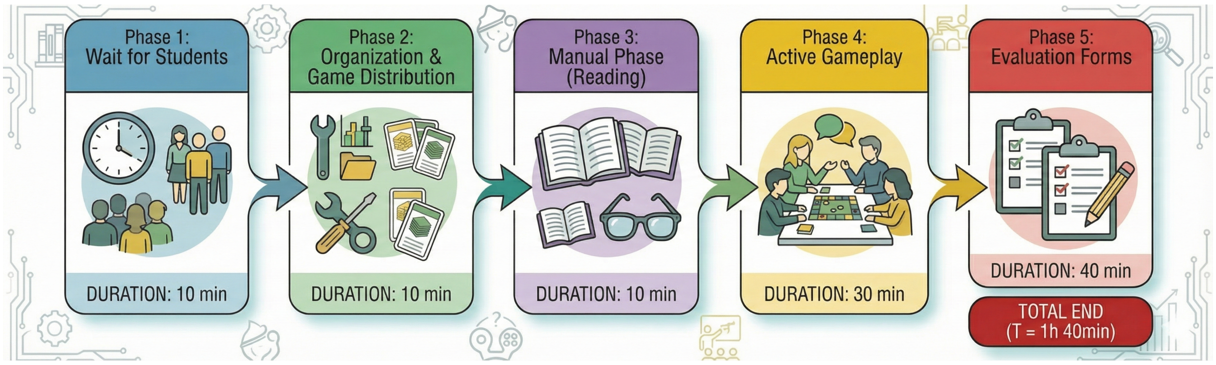


Figure 5.1: Flow of the evaluation application.

The gameplay session lasted approximately 30 minutes. Due to the self-paced nature of the groups, the number of completed rounds varied, with an estimated average of 8 rounds per group. Since the role of Architecture Master (AM) rotates after every round, this volume ensured that the AM role was distributed among participants, allowing most students to experience both the investigator and facilitator perspectives. Throughout the session, the researcher remained present but intentionally limited interventions. Assistance was restricted to clarifying operational doubts regarding rules, avoiding any interference in the students' investigative reasoning or decision-making processes.

Immediately following the gameplay, data were collected using an online questionnaire (Google Forms). Ethical considerations were strictly observed throughout the process; participants were only allowed to proceed to the questions after reviewing and accepting the Informed Consent Form (ICF) provided in Appendix C.

To protect participant privacy, the survey was designed to be entirely anonymous, ensuring that no sensitive personal information, such as names, student IDs, or contact details, was collected or stored. Furthermore, participation was strictly voluntary, and students were informed of their right to withdraw at any stage without any prejudice or penalty. To ensure data integrity and reduce social desirability bias, the form was configured to block multiple submissions from the same device. Quantitative data were exported for statistical analysis, while qualitative responses were processed to identify recurring perceptions and opportunities for improvement, maintaining the confidentiality of individual inputs at all times.

5.3 Results and Data Analysis

This section presents an extended and rigorous analysis of the data collected from the 34 participants ($N = 34$) who evaluated Architectural Stories using the MEEGA+, IMI, and IMMS instruments. All students' responses to the questionnaires are publicly available for inspection and replication purposes[16]. The analysis integrates descriptive statistics, internal consistency indicators, and qualitative feedback, providing a comprehensive perspective on the pedagogical value and experiential quality of the game.

5.3.1 Participants' Profile

A total of 34 students participated in the evaluation. The majority were young adults, with 30 participants (88.2%) aged 18–28 years, followed by 3 participants (8.8%) in the 29–39 age group, and 1 participant (2.9%) aged 40–50 years. This distribution indicates that the sample is strongly concentrated in younger age ranges.

In terms of gender identity, the sample consisted of 20 male participants (58.8%), 13 female participants (38.2%), and one non-binary participant (2.9%). Although the distribution is not balanced, it still reflects the presence of diverse gender identities.

Regarding the frequency of playing non-digital games, the data revealed diverse levels of participation. Specifically, 18 participants (52.9%) stated that they play rarely, while 12 participants (35.3%) reported playing monthly. Only 2 participants (5.9%) indicated playing weekly, one participant (2.9%) reported playing daily, and one participant (2.9%) stated that they never engage in non-digital games.

[16]https://tinyurl.com/4vbwb3ts

5.3.2 Results Obtained from the Application of MEEGA+

To assess the perceived quality of Architectural Stories, the MEEGA+ (Annex B) was employed as one of the evaluation instruments. This model provides a comprehensive analysis by decomposing quality into two major factors: Usability and Player Experience. The instrument used in this study consists of 32 quantitative items, measured on a 5-point Likert scale, and is complemented by three open-ended questions designed to capture subjective feedback regarding positive aspects, negative aspects, and suggestions for improvement. These qualitative responses were analyzed to corroborate the quantitative data, providing contextual nuance to the statistical findings presented in this section.

Figure 5.2 details the distribution of responses across all MEEGA+ items. Visual inspection enables the immediate identification of the game's strongest aspects, as indicated by the predominance of dark blue bars (Strongly Agree). Notably, the items related to Social Interaction and Relevance achieved near-unanimous consensus. For instance, the statement "*It is clear to me how the contents of the game are related to the course*" reached approximately 91% of Strongly Agree responses, confirming that students clearly perceived the educational purpose of the activity. Similarly, the visual design was highly praised, with 88% of participants strongly agreeing that "*The colors used in the game are meaningful*".

Conversely, the chart explains the lower mean observed in the Focused Attention dimension. The item "*I forgot about my immediate surroundings while playing this game*" exhibits the most significant divergence, with a visible portion of disagreement (red and pink bars totaling approximately 27%). This result is expected for a classroom activity based on loud negotiation and group debate, where isolation from the environment is neither achieved nor intended.

Table 5.1 summarizes the descriptive statistics calculated from participant scores. The consistently high mean values across dimensions, many exceeding 4.5 on a 5-point Likert scale, demonstrate the strong acceptance and effectiveness of the game.

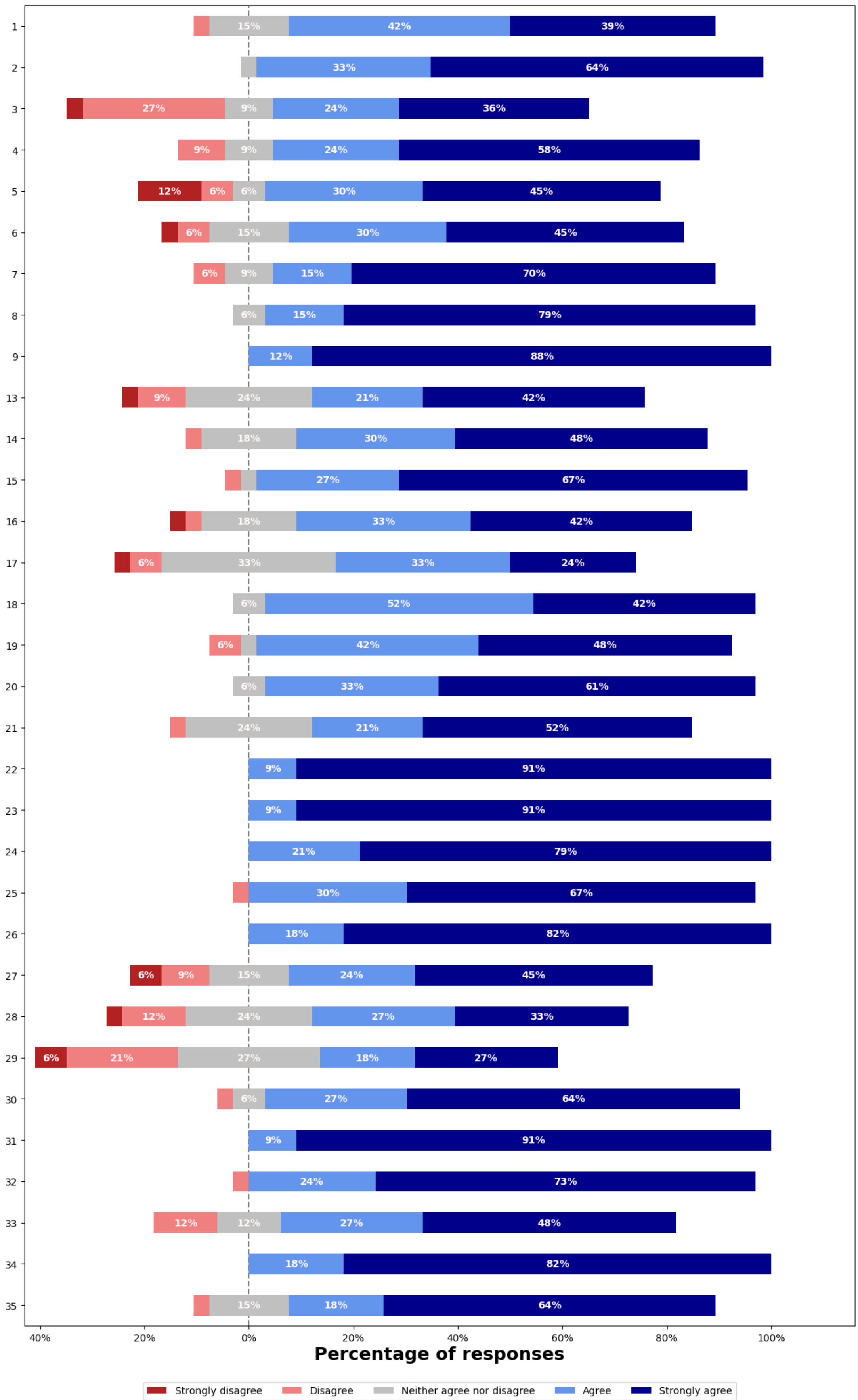


Figure 5.2: Distribution of responses across MEEGA+ items.

Table 5.1: Descriptive Statistics of MEEGA+ Dimensions ($N = 34$).

Factor	Subdimension	Items	Mean (μ)	Std. Dev. (σ)
Usability	Aesthetics	1–2	4.40	0.72
	Learnability	3–5	3.94	1.24
	Operability	6–7	4.29	0.99
	Accessibility	8–9	**4.81**	0.47
Player Experience	Confidence	13–14	4.07	0.47
	Challenge	15–17	4.12	0.98
	Satisfaction	18–21	4.36	0.76
	Social Interaction	22–24	**4.87**	0.34
	Fun	25–26	4.72	0.54
	Focused Attention	27–29	3.69	1.22
	Relevance	30–33	4.56	0.78
	Perceived Learning	34–35	4.63	0.69

Usability: Strong Aesthetic Appeal and Exceptional Accessibility

Usability scores reveal a highly positive perception of the game. The Aesthetics, with a mean of ($\mu = 4.40$) and a relatively low standard deviation ($\sigma = 0.72$), indicate a broad consensus that the visual design (colors, symmetry, and consistency) met high-quality expectations. This aligns with multiple qualitative comments praising the attractive board, clear typography, and well-designed visual identity.

The strongest subdimension in the entire Usability factor was Accessibility ($\mu = 4.81$, $\sigma = 0.47$), with Item 9 (*"The colors used are comprehensible"*) reaching an exceptionally high mean of 4.88. This confirms that the visual environment effectively supports comprehension, even for students unfamiliar with tabletop mechanics.

This high accessibility score is particularly significant when cross-referenced with the participants' gaming habits reported in Section 5.3.1. Given that 52.9% of the students stated they play non-digital games "rarely", the strong positive evaluation indicates that the game's design successfully mitigates entry barriers. It demonstrates that the visual aids and mechanics are intuitive enough to engage a general audience, ensuring that the lack of prior gaming literacy does not hinder the educational experience.

Conversely, Learnability ($\mu = 3.94$, $\sigma = 1.24$) displayed the highest variability in the entire questionnaire. This deviation reflects the heterogeneous familiarity with Architectural Patterns among students. For learners with limited domain knowledge, the text-heavy descriptions introduced a cognitive barrier. As the participant P9 noted:

> *"The role of the patterns could be better structured, with images... to help those who do not know the patterns." (P9)*

Such insights reinforce that future iterations of the game should include visual scaffolding (e.g., diagrams, icons, and simplified pattern summaries). These refinements would reduce extraneous cognitive load and promote quicker onboarding.

Player Experience: Engagement, Collaboration, and Positive Learning Dynamics

The Player Experience results offer the strongest evidence of the effectiveness of Architectural Stories. All subdimensions scored above 4.0, with the exception of Focused Attention, a finding that aligns with the nature of socially driven learning activities.

Social interaction emerged as the primary driver of engagement. Social Interaction was the highest-rated subdimension across all MEEGA+ metrics ($\mu = 4.87$, $\sigma = 0.34$), reflecting an almost unanimous perception that the game successfully promotes collaborative reasoning, debate, and collective meaning making, which are central elements of social constructivist learning.

> *" I liked the interactivity with the other classmates that the game promotes..." (P9)*

The results also highlight strong affective engagement, as evidenced by high scores in Fun ($\mu = 4.72$) and Satisfaction ($\mu = 4.36$). These values suggest that the balance between challenge and reward was well-calibrated, enabling players to experience enjoyment while progressing through the activity. Student comments further reinforce this positive reception. When asked "What did you like most about the game?", they responded with comments such as:

> *"Reminded me of Black Stories. I found it really fun." (P19)*

> *"Learning while playing." (P14)*

Although Focused Attention obtained a comparatively lower mean ($\mu = 3.69$) and a higher standard deviation ($\sigma = 1.22$), this should not be interpreted as a shortcoming. Because the game is intentionally designed to stimulate discussion, negotiation, and shared analysis, the classroom naturally becomes more dynamic. In this context, attention is distributed across peers and collective interactions, rather than anchored in solitary, introspective concentration.

Finally, the dimensions Challenge ($\mu = 4.12$) and Confidence ($\mu = 4.07$) form a coherent pair, suggesting that participants perceived the tasks as appropriately demanding yet manageable. Several respondents emphasized that solving the scenarios and uncovering Architectural Patterns provided them with a clear sense of accomplishment, indicating that the game fosters both cognitive stimulation and self-efficacy.

Perceived Learning and Educational Value

The most significant pedagogical outcomes appear in the Relevance ($\mu = 4.56$) and Perceived Learning ($\mu = 4.63$) scores. These metrics confirm that the game did not merely entertain; it effectively reinforced the key concepts of SA.

The item *"The game contributed to my learning in the course"* achieved one of the highest mean scores in the dataset ($\mu = 4.82$), reflecting a strong consensus among participants that the activity effectively supported their learning process.

In qualitative responses, students repeatedly emphasized how the game facilitated understanding:

> *"I was able to learn concepts that I didn't know before."* (P25)
> *"The reference sheet of the 'Patterns' is a direct and efficient summary of the content, and interacting with it helped reinforce the concepts."* (P6)

These testimonials, combined with the quantitative data, confirm that the game successfully bridged the gap between abstract theory and practical understanding, validating its educational value.

5.3.3 Results Obtained from the Application of IMI

To evaluate the participants' intrinsic motivation regarding the educational game Architectural Stories, IMI was employed. The instrument comprises 37 items distributed across six subscales, rated on a 7-point Likert scale ranging from 1 (Strongly Disagree) to

7 (Strongly Agree). The complete questionnaire containing all items used in this study is available in Annex C.

Scoring Methodology and Reverse Items

To ensure the reliability of the data and control for acquiescence bias (the tendency of respondents to agree with statements regardless of content), the questionnaire includes both positive and negative statements. Items marked with (R) are reverse-scored. For example, in the Interest/Enjoyment subscale, while item 1 is positive ("I enjoyed doing this activity very much"), item 3 is negative ("I thought this was a boring activity").

Before calculating the mean scores for each factor, the responses to these reverse items were inverted using the standard formula for Likert scales:

$$Score_{adjusted} = (k + 1) - Score_{raw} \tag{5.1}$$

Where k is the maximum value of the scale (7). Therefore, the formula used was $Score_{adjusted} = 8 - Score_{raw}$. This transformation ensures that a "Strongly Disagree" (1) response to a negative statement (e.g., "I felt pressured") is mathematically equivalent to a "Strongly Agree" (7) response to a positive motivation statement, allowing for consistent aggregation within the subscales.

Data Analysis

Figure 5.3 illustrates the distribution of responses for all 37 items using a diverging stacked bar chart. This visualization highlights the frequency of agreement (blue hues) versus disagreement (red hues) after the necessary inversions, providing a granular view of the participants' feedback.

Table 5.2 summarizes the descriptive statistics (Mean and Standard Deviation) for each of the six IMI factors ($N = 34$). The results are analyzed in detail below, linking quantitative data to the theoretical constructs of Self-Determination Theory.

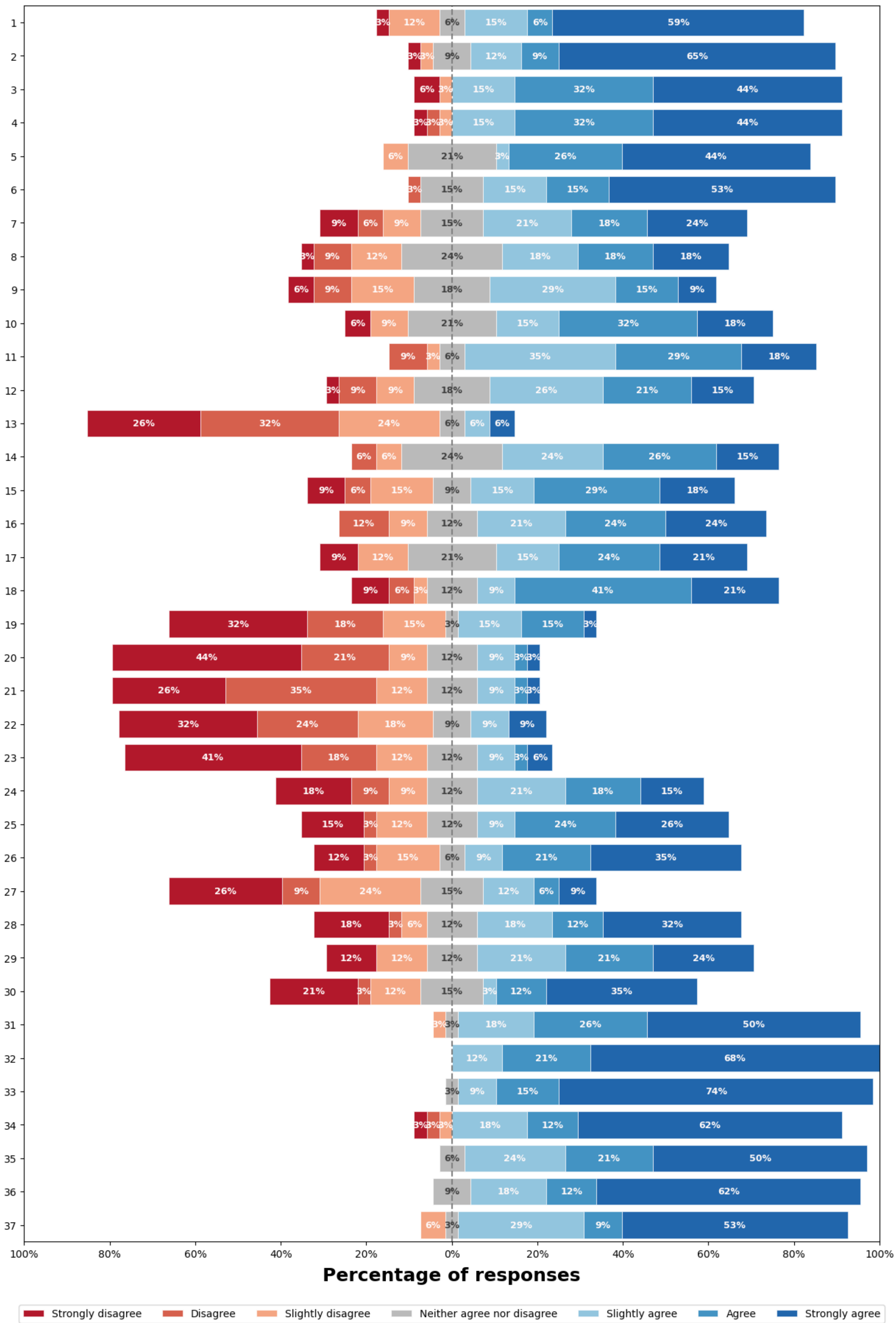


Figure 5.3: Distribution of responses for IMI items.

Table 5.2: Descriptive Statistics of IMI Factors ($N = 34$).

Factor	**Items**	**Mean (μ)**	**Std. Dev. (σ)**
Interest/Enjoyment	1–7	5.77	1.58
Perceived Competence	8–13	4.93	1.61
Effort/Importance	14–18	4.96	1.72
Pressure/Tension	19–23	2.69	1.80
Perceived Choice	24–30	4.48	2.16
Value/Usefulness	31–37	**6.26**	1.08

Interest/Enjoyment

According to the IMI definitions, the Interest/Enjoyment subscale is considered the self-report measure of intrinsic motivation per se. While other factors serve as antecedents, this subscale directly assesses the participant's enjoyment of the activity. The obtained mean of $\mu = 5.77$ indicates a high level of engagement. Notably, items such as "I enjoyed doing this activity very much" (Q1) and "This activity was fun to do" (Q2) received predominantly positive responses. This result confirms that the gamification elements successfully transformed the study of software patterns into an engaging experience, preventing the boredom often associated with theoretical lectures.

Perceived Competence

This subscale posits that intrinsic motivation is strengthened when individuals perceive themselves as effective at an activity. The mean score of $\mu = 4.93$ reflects a moderate-to-high sense of competence. Participants felt reasonably skilled at the game, though the standard deviation ($\sigma = 1.61$) suggests some variation. This variance is expected in educational games: if the game is too easy, competence is high but interest drops; if too hard, competence drops. The obtained score suggests a "flow" state balance where the challenge was significant enough to require effort but not so difficult as to make students feel incapable.

Effort/Importance

The Effort/Importance subscale measures the amount of energy participants perceive they invested in the activity. The mean score ($\mu = 4.96$) indicates that students did not play passively; they mobilized cognitive resources to solve the architectural puzzles. In the context of learning, a high level of effort combined with high interest is the ideal scenario for active learning. It suggests that students were trying hard, not because they were forced, but because they valued the outcome of the game.

Pressure/Tension

Pressure/Tension is a negative predictor of intrinsic motivation. Unlike the other subscales, higher scores on this subscale would indicate anxiety, which undermines the learning experience. The results were highly positive, with a low mean of $\mu = 2.69$. As shown in Figure 5.3 (items 19–23), the vast majority of responses indicate a relaxed state. This demonstrates that the game environment was psychologically safe. Students did not feel nervous or pressured to perform, which fosters a mindset conducive to exploration and error-based learning without fear of judgment.

Perceived Choice

This subscale assesses the participants' sense of autonomy. The results here were the most mixed, with the lowest positive mean ($\mu = 4.48$) and the highest variability ($\sigma = 2.16$). The high standard deviation implies a strong divergence in user experience. The presence of reverse items, such as "I felt like I had to do this" (Q27), highlights that while some students felt they participated by choice, others may have perceived the mandatory nature of the classroom activity as a constraint on their autonomy. This suggests a potential area for improvement in future iterations, perhaps by allowing more flexible rules or optional paths within the game to enhance the sense of volition.

Value/Usefulness

Finally, the Value/Usefulness subscale embodies the idea that people internalize activities they experience as valuable for themselves. This factor achieved the highest mean ($\mu = 6.26$) and the lowest standard deviation ($\sigma = 1.08$), representing the strongest finding of this study. The items Q31–Q37 show an almost unanimous consensus. This confirms that students clearly recognized the pedagogical value of "Architectural Stories." Regardless of whether they found the game easy or hard, they agreed it was highly relevant for improving their understanding of SA content.

5.3.4 Results Obtained from the Application of IMMS

The IMMS was employed to evaluate the motivational quality of the instructional components of the game Architectural Stories. The IMMS is theoretically grounded in the ARCS motivational model, which defines four core dimensions that support learner motivation: Attention, Relevance, Confidence, and Satisfaction. The instrument is widely used in educational research to assess learners' perceptions of instructional design and motivational appeal.

Scoring Methodology and Reverse Items

The IMMS consists of 36 items evaluated on a five-point Likert scale, ranging from 1 (Not true) to 5 (Very true). To minimize acquiescence bias and improve measurement reliability, the questionnaire includes several reverse-worded items, identified with an R. Before statistical aggregation, these items were reverse-scored using the standard transformation for five-point scales:

$$Score_{adjusted} = 6 - Score_{raw} \tag{5.2}$$

This procedure ensures that all items contribute consistently to their respective motivational constructs, such that higher scores uniformly represent more positive motivational perceptions. The complete questionnaire, including item allocation to ARCS factors and the identification of reverse-scored items, is provided in Annex D.

Overview of Results

Figure 5.4 presents the distribution of responses across all questionnaire items, while Table 5.3 summarizes the descriptive statistics for the four motivational dimensions defined by the ARCS model. Across all factors, mean scores were above the midpoint of the scale, indicating an overall positive motivational evaluation of the instructional game.

The graphical distribution reveals a strong predominance of positive responses, with most items showing high concentrations in the categories 'Agree' and 'Strongly

Agree'. Negative responses were infrequent and generally limited to small proportions of participants, suggesting that motivational challenges, when present, were localized and did not compromise the overall instructional experience.

Table 5.3: Descriptive Statistics of IMMS Factors ($N = 34$).

Factor	**Mean (μ)**	**Std. Dev. (σ)**
Attention (A)	3.90	1.08
Relevance (R)	4.03	1.14
Confidence (C)	3.74	1.18
Satisfaction (S)	**4.18**	0.94

Attention

The Attention dimension assesses the extent to which the instructional material captures and sustains learners' curiosity and interest. This factor achieved a mean score of $\mu = 3.90$, indicating a strong level of engagement. The response distribution indicates a predominance of agreement, with relatively few negative responses, suggesting that most participants found the game engaging.

According to the ARCS model, attention is supported through perceptual and inquiry-based arousal. The visual presentation, narrative structure, and interactive challenges of Architectural Stories appear to have contributed to sustained engagement, even when learners were confronted with abstract and conceptually demanding topics such as SA patterns. Some neutral responses observed in this dimension suggest individual variation in sustained attention, which is expected in heterogeneous learning groups.

Relevance

The Relevance dimension evaluates the perceived alignment between the instructional content and learners' academic or professional goals. This factor obtained a mean score of $\mu = 4.03$, reflecting a strong perception of applicability. The graphical distribution shows a clear concentration of responses in the 'Agree' and 'Strongly Agree' categories, with minimal disagreement.

These results indicate that participants perceived the game content as meaningful and closely related to real-world software engineering practices. The contextualized archi-

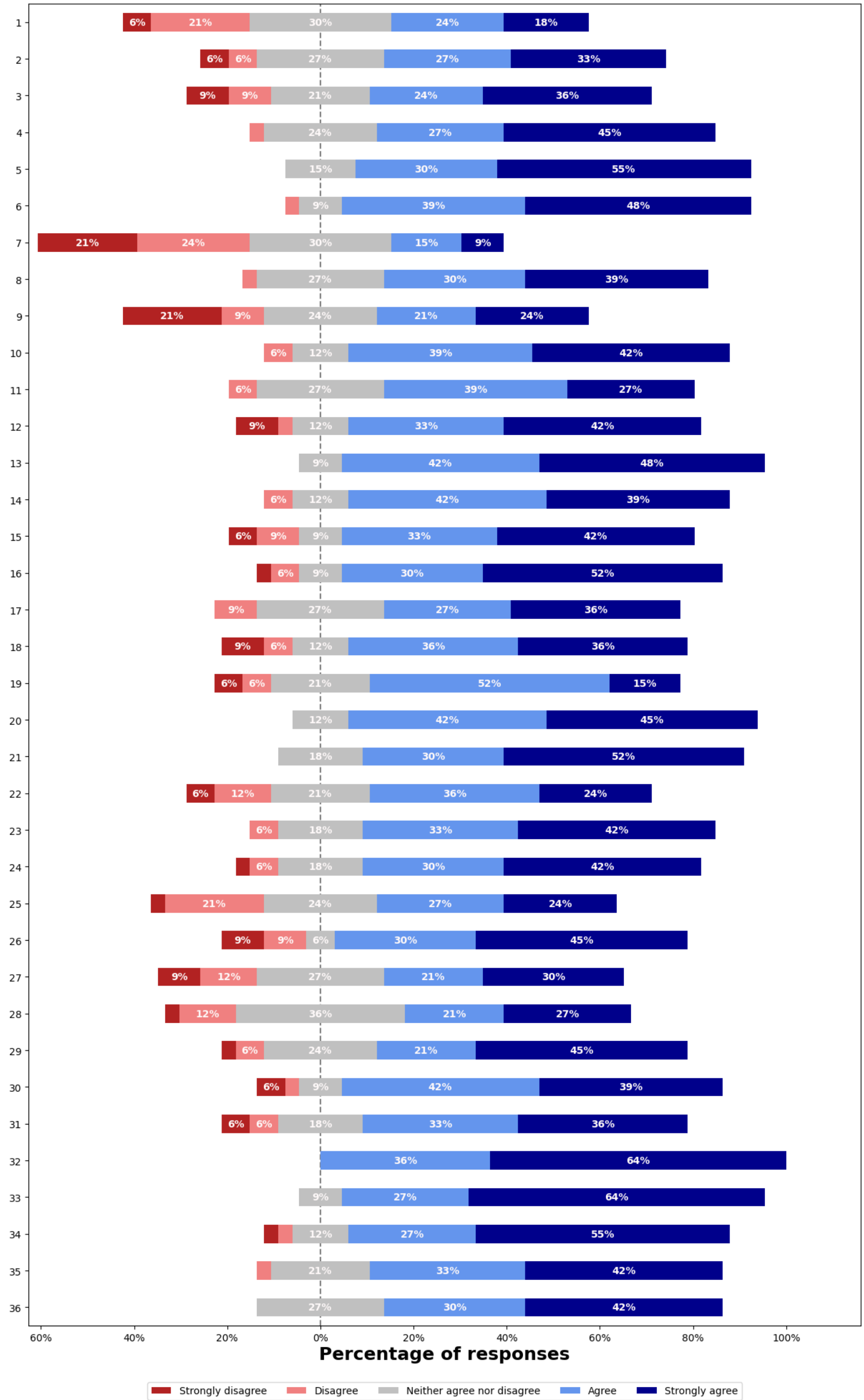


Figure 5.4: Distribution of responses for IMMS items.

tectural scenarios and decision-making processes reinforced the usefulness of the instructional material, a central condition for sustained motivation within the ARCS framework.

Confidence

The Confidence dimension measures learners' expectations for success and their perceived control over learning outcomes. This factor presented a mean score of $\mu = 3.74$, which, although positive, was lower than those observed for Relevance and Satisfaction. The response distribution reveals a greater presence of neutral responses, suggesting variability in learners' confidence levels.

From the perspective of the ARCS model, this pattern indicates that while the game provided sufficient support for most participants, some learners experienced uncertainty when engaging with complex architectural concepts. Nonetheless, the limited occurrence of negative responses suggests that the balance between challenge and instructional support was generally effective and did not lead to widespread frustration.

Satisfaction

Satisfaction refers to the positive affective responses associated with learning achievements and the instructional experience. This dimension achieved the highest mean score among all factors ($\mu = 4.18$) and exhibited the strongest concentration of Strongly agree responses in the graphical distribution.

These results indicate that learners experienced a high level of enjoyment and a sense of accomplishment when interacting with the game. According to the ARCS framework, satisfaction plays a critical role in reinforcing motivation and promoting positive attitudes toward future learning activities. The high satisfaction scores suggest that the instructional design of Architectural Stories was effective not only in supporting learning outcomes but also in fostering a rewarding educational experience.

Overall, the IMMS results provide consistent evidence, at both the statistical and item-level distributions, that the instructional design of Architectural Stories successfully supported learner motivation in alignment with the principles of the ARCS model.

5.3.5 Discussion

This work investigated the motivational and educational impact of the Architectural Stories, guided by the RQ presented in Section 1.2. To address this question comprehensively, the game was evaluated using three complementary instruments: MEEGA+, IMI, and IMMS, each targeting distinct yet interrelated dimensions of the learning experience. The triangulation of results provides consistent and robust evidence that the card-based game has a positive influence on students' motivation, engagement, and perceived learning of software architectural patterns.

From the perspective of game quality and player experience, the MEEGA+ results indicate a high level of acceptance and effectiveness. The strong performance across both major factors, Usability and Player Experience, demonstrates that the game successfully balances clarity of interaction with engaging gameplay. In particular, the exceptionally high scores in Accessibility, Social Interaction, and Relevance highlight that the design choices adopted in Architectural Stories effectively lowered entry barriers, even for participants with limited prior experience with non-digital games. This is a relevant finding in an educational context, as more than half of the participants reported playing tabletop games rarely. Despite this, the game remained intuitive, visually comprehensible, and easy to engage with, reinforcing its inclusiveness as a pedagogical tool.

The prominence of Social Interaction as the highest-rated MEEGA+ dimension is significant. It suggests that motivation and engagement emerged primarily through collaborative reasoning, debate, and collective problem solving, rather than through individual immersion. This interpretation is further supported by the lower scores in Focused Attention, which should not be understood as a weakness of the design. Instead, they reflect the inherently social and dialogical nature of the activity, where attention is distributed across peers, arguments, and shared decision-making processes. In this sense, the game aligns with social constructivist perspectives on learning, which hold that knowledge is actively constructed through the interaction and negotiation of meaning.

The IMI results complement the MEEGA+ findings by providing deeper insight into the students' intrinsic motivational states. High scores in Interest/Enjoyment confirm that the card game format transformed the learning of architectural patterns into an engaging and enjoyable experience, mitigating the abstraction and cognitive heaviness typically associated with this topic. More importantly, the Value/Usefulness subscale achieved the highest mean across all IMI factors, indicating a strong internalization of the educational value of the activity. This result is central to answering the research question, as it demonstrates that motivation was not driven solely by fun or novelty, but by a clear recognition of the game's contribution to learning.

Moderate-to-high scores in Perceived Competence and Effort/Importance further indicate that students felt capable of engaging with the challenges presented by the game and were willing to invest effort in solving them. The observed variability in these dimensions reflects the heterogeneous background knowledge of the participants, particularly in terms of their familiarity with architectural patterns. However, this variability did not translate into increased anxiety, as evidenced by the low scores on the Pressure/Tension scale. This combination of challenge, effort, and low pressure suggests that the game fostered a psychologically safe learning environment, in which students could explore ideas, make mistakes, and learn collaboratively without fear of negative evaluation.

The Perceived Choice subscale presented the most divergent responses, indicating that some students experienced limited autonomy due to the mandatory nature of the classroom activity. This finding highlights an inherent tension in formal educational settings, where instructional activities are often required rather than voluntary. Nevertheless, even in the presence of reduced perceived autonomy for some participants, the overall motivational profile remained strongly positive. This suggests that the game's relevance, enjoyment, and collaborative structure were sufficient to sustain motivation despite contextual constraints.

The IMMS results provide additional confirmation from an instructional design perspective, grounded in the ARCS motivational model. High scores in Attention indicate that the narrative structure, visual elements, and interactive challenges of Architectural Stories successfully captured and maintained learners' interest. The Relevance dimension reinforces earlier findings from both MEEGA+ and IMI, showing that students consistently perceived the instructional content as aligned with their academic goals and future professional practice in software engineering.

Although the Confidence dimension obtained slightly lower mean values compared to other IMMS factors, this result does not undermine the overall effectiveness of the instructional design. Instead, it highlights opportunities for refinement, particularly in supporting learners with less prior knowledge of architectural patterns. The inclusion of additional visual scaffolding, simplified summaries, or progressive disclosure of complex information could further enhance learners' confidence without reducing the intellectual challenge of the activity. Importantly, the high scores in Satisfaction indicate that students experienced a strong sense of accomplishment and positive affect, reinforcing motivation and supporting favorable attitudes toward future learning activities.

When considered together, the findings from MEEGA+, IMI, and IMMS offer a coherent and multifaceted answer to the research question posed in Section 1.2. The use of a card game influences students' motivation to learn software architectural patterns by activating multiple motivational mechanisms simultaneously. First, it increases situational interest through playful and interactive dynamics. Second, it fosters collaborative learning by structuring meaningful social interaction and collective reasoning. Third, it supports the internalization of learning value by clearly connecting game activities to course content and professional relevance. These mechanisms operate in synergy, resulting in heightened engagement, sustained effort, and positive perceptions of the learning experience.

In summary, the discussion demonstrates that Architectural Stories functions not merely as a gamified supplement to traditional instruction but as a mediating educational artifact that bridges abstract theoretical knowledge and collaborative sense-making. The consistency of positive results across three well-established evaluation instruments strengthens the validity of this conclusion and underscores the potential of card-based educational games as effective pedagogical strategies in software engineering education. The findings also provide clear directions for future improvements and research, particularly regarding adaptive scaffolding and autonomy support, further consolidating the contribution of this work.

5.4 Threats to Validity in the Evaluation of Game

Despite the efforts made to ensure methodological rigor in the evaluation of Architectural Stories, it is necessary to acknowledge that certain threats to validity may have influenced the findings. Explicitly identifying these threats enables a more careful interpretation of the results and their limitations (MOURÃO et al., 2020). The main concerns and their respective mitigation strategies are discussed below.

A primary threat to **external validity** concerns the generalizability of the results. The evaluation involved 34 students enrolled in a single Software Engineering course at one university. This sample may not adequately represent students from other institutions, learners with different academic backgrounds, or practitioners in professional settings. Additionally, participation occurred within a formal classroom context, which may influence motivation and engagement differently from informal or voluntary learning environments. Replications with larger and more diverse populations, as well as applications in different educational and professional contexts, are necessary to strengthen the generalizability of the findings.

Threats to **construct validity** relate to whether the instruments used accurately captured the intended constructs, such as engagement, motivation, perceived learning, and instructional quality. The study relied primarily on self-reported data collected through three validated instruments: MEEGA+, IMI, and IMMS. Although these instruments are well established and widely used, they assess participants' perceptions rather than objective learning outcomes. As a result, constructs such as learning effectiveness and competence were measured subjectively, without direct verification through performance-based assessments, such as pre- and post-tests. Furthermore, the novelty of the game-based activity may have positively influenced responses across all three instruments, particularly in dimensions related to enjoyment, interest, and attention.

Regarding **internal validity**, the short duration of the gameplay session, approximately 30 minutes, limits the ability to evaluate sustained engagement or long-term learning effects. The instructional material and game rules were provided for autonomous group study, which may have led to variations in interpretation and gameplay dynamics across groups. These differences could have influenced both the learning experience and the motivational responses captured by the MEEGA+, IMI, and IMMS. Additionally, the absence of a control group using a traditional instructional approach prevents causal inferences about the effectiveness of the game compared to alternative teaching strategies.

Finally, threats to **conclusion validity** from the analytical approach adopted in the study. The results were analyzed using descriptive statistics, including means, frequencies, and standard deviations, without inferential statistical testing. Consequently, it is not possible to determine whether observed differences between dimensions or instruments are statistically significant. The relatively small sample size also limits statistical power and increases sensitivity to individual variation. While the triangulation of MEEGA+, IMI, and IMMS strengthens interpretive confidence by providing complementary perspectives, the findings should still be interpreted as exploratory and indicative of perceived experience rather than as evidence of statistically robust causal relationships.

Overall, while these threats do not invalidate the results, they highlight limitations and point to clear opportunities for methodological refinement in future studies.

5.5 Final Considerations

The evaluation results presented in this chapter consistently demonstrate that Architectural Stories is an effective and motivating tool for teaching software architectural patterns. Thedata from MEEGA+, IMI, and IMMS confirmed that the game not only promotes high social interaction and engagement but also successfully bridges the gap between abstract concepts and practical understanding. While the analysis identified opportunities for refinement in scaffolding for novice learners, the overall positive impact on student motivation and perceived learning validates the research-driven design of the artifact. With the empirical validation complete, the following chapter provides the conclusions of this work, summarizing its contributions, presenting the publications derived from this research, and outlining directions for future studies to advance SAE further.

6 Conclusion

SAE requires supporting students in reasoning about design, connecting contexts to solutions, and articulating architectural rationale. Although literature reports several game-based approaches to foster engagement, few focus on training learners to interpret narrative descriptions and infer Architectural Patterns from contextual clues. Architectural Stories addresses this gap by combining narrative deduction, collaborative reasoning, and pattern-based design.

This work proposes Architectural Stories, a non-digital educational card game where students collaboratively analyze architectural scenarios and construct solutions integrating diagnosis, justification, and prescription. The evaluation with 34 undergraduates combined three instruments (MEEGA+, IMI, and IMMS), providing a multidimensional perspective on usability, experience, motivation, and learning. Results indicate strong acceptance, high engagement, and positive perceptions regarding relevance, learning support, and instructional value.

6.1 Contributions

The **first contribution** of this work is a comprehensive SMS that synthesizes two decades of research (2005–2025) on Software Architecture Education. By analyzing 45 primary studies, this mapping provides the most up-to-date and extensive overview of the field to date, identifying dominant topics, educational strategies, tools, and empirical evidence. Beyond consolidating existing knowledge, the SMS highlights persistent gaps, such as the limited focus on narrative reasoning, the challenge of bridging abstraction and practice, and the need for approaches that better simulate real-world architectural ambiguity. This contribution establishes a rigorous foundation for both researchers and educators, directly informing the design decisions of the educational intervention proposed in this work.

A **second contribution** of this work is the proposal of an educational game that emphasizes the interpretation of narrative scenarios as a core architectural competency.

Unlike approaches that focus on pattern recognition or classification, Architectural Stories requires players to reason from incomplete and contextualized descriptions, simulating the ambiguity and trade-off analysis inherent in real-world architectural decision-making.

A **third contribution** lies in the development of a reusable and well-structured educational package, including rulebooks, scenario cards, facilitator guidelines, and reference materials on Architectural Patterns. This package enables instructors to integrate the game into different moments of a course, such as introductory activities, consolidation exercises, or collaborative reviews, without requiring extensive prior preparation.

The **fourth contribution** is the empirical evaluation of the game using validated instruments. The combined use of MEEGA+, IMI, and IMMS allowed for triangulation of results related to usability, player experience, intrinsic motivation, and instructional quality. The findings demonstrate that the game supports collaborative learning, promotes perceived learning and relevance, and fosters intrinsic motivation, even among students with limited prior experience in non-digital games.

In summary, Architectural Stories offers an effective approach to teaching Architectural Pattern selection through narrative interpretation and collaborative reasoning. By addressing a higher-order cognitive skill that is often challenging to develop through traditional instruction, the game contributes to both research and practice in SAE. With continued refinement, expanded evaluation, and the development of a digital version, Architectural Stories has strong potential to become a versatile instructional tool for academic and professional learning contexts.

6.2 Publications

The research conducted in this work resulted in the following publications:

1. MENEZES, Maria Clara Ribeiro de; VALLE, Pedro Henrique Dias; OLIVEIRA, Alessandreia Marta. Two Decades of Software Architecture Education: State of the Art, Challenges, and Future Directions. In: *ICSE-SEET '26. 2026 IEEE/ACM 48th International Conference on Software Engineering, Software Engineering Education and Training Track*, April 12–18, 2026, Rio de Janeiro, Brazil. DOI: ⟨https://dx.

doi.org/10.1145/3786580.3786954⟩. (Accepted for publication)

2. MENEZES, Maria Clara Ribeiro de; VALLE, Pedro Henrique Dias; OLIVEIRA, Alessandreia Marta. A Card Game for Architectural Patterns Education. Submitted to: *EduComp 2026 Brazilian Symp. on Computing Education.* (Under review)

6.3 Future Work

Several directions for future work emerge from both the quantitative results and the qualitative feedback collected during the evaluation. One immediate improvement concerns enhancing learnability through the addition of visual scaffolding. Participants suggested the inclusion of diagrams, icons, and more concise summaries of Architectural Patterns to reduce cognitive load, particularly for students with less prior exposure to the topic.

Another direction involves broadening the empirical evaluation. Future studies should replicate the game with more diverse audiences, including early-stage undergraduates, graduate students, and industry practitioners, as well as in different institutional contexts. More rigorous experimental designs, incorporating pre- and post-tests and control groups, would enable the assessment of learning gains beyond self-reported perceptions.

Comparative studies with other instructional strategies, such as case-based discussions, existing architecture games, or traditional exercises, also represent a promising avenue to position Architectural Stories within the broader landscape of SAE.

Finally, we will focus on refining and systematically evaluating the digital version of Architectural Stories. This digital version is currently under development and has an initial functional release available online[17]. Future efforts will focus on enhancing usability, gameplay balance, and pedagogical alignment, as well as conducting empirical studies to evaluate the software's effectiveness in supporting software architecture learning in online and hybrid settings. Additionally, the digital format opens opportunities for features such as automated feedback, adaptive difficulty, learning analytics, and support for larger groups, while also helping to reduce variability in rule interpretation and gameplay flow observed during presencial gameplay sessions.

[17]https://mariaclara11.github.io/architectural-stories/

Bibliography

AL-QORA'N, L. F.; JAWARNEH, A.; NGANJI, J. T. Toward creating software architects using mobile project-based learning model (mobile-pbl) for teaching software architecture. *Multimodal Technologies and Interaction*, MDPI, v. 7, n. 3, p. 31, 2023.

ALPEROWITZ, L.; JOHANSSEN, J. O.; DZVONYAR, D.; BRUEGGE, B. Modeling in agile project courses. In: *MODELS (Satellite Events)*. [S.l.: s.n.], 2017. p. 521–524.

ANGELOV, S.; BEER, P. de. An approach to software architecting in agile software development projects in education. In: SPRINGER. *European Conference on Software Architecture*. [S.l.], 2015. p. 157–168.

ARAÚJO, A.; COELHO, A.; RODRIGUES, M. E.; VIANA, W.; MARQUES, A. B. Todo esforço será recompensado: Gamificaçao no ensino de arquitetura de software com o uso de badges. In: SBC. *Workshop sobre Educação em Computação (WEI)*. [S.l.], 2024. p. 341–352.

BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. [S.l.]: Addison-Wesley Professional, 2021.

BOER, R. C. D.; LAGO, P.; VERDECCHIA, R.; KRUCHTEN, P. Decidarch v2: An improved game to teach architecture design decision making. In: IEEE. *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*. [S.l.], 2019. p. 153–157.

BOER, R. C. de; FARENHORST, R.; VLIET, H. van. A community of learners approach to software architecture education. In: IEEE. *2009 22nd Conference on Software Engineering Education and Training*. [S.l.], 2009. p. 190–197.

CAO, M.; CAO, Z. Teaching data structures and software architecture while constructing curriculum platform. In: IEEE. *2011 6th International Conference on Computer Science & Education (ICCSE)*. [S.l.], 2011. p. 1433–1437.

CARVALHO, E. A.; BARBOSA, J. R.; NETO, R. B.; NETO, V. V. G.; VALLE, P. H. D. Decora: um sistema de apoio ao ensino de decisões de projetos arquiteturais. *Anais*, p. 719–729, 2025.

CASTRO, L. M. Role-playing software architecture styles. In: IEEE. *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*. [S.l.], 2023. p. 171–174.

CERVANTES, H.; HAZIYEV, S.; HRYTSAY, O.; KAZMAN, R. Smart decisions: an architectural design game. In: *Proceedings of the 38th International Conference on Software Engineering Companion*. [S.l.: s.n.], 2016. p. 327–335.

CHENOWETH, S.; ARDIS, M.; DUGAS, C. Adapting cooperative learning to teach software architecture in multiple-role teams. 2007.

CONNOLLY, T. M.; BOYLE, E. A.; MACARTHUR, E.; HAINEY, T.; BOYLE, J. M. A systematic literature review of empirical evidence on computer games and serious games. *Computers & education*, Elsevier, v. 59, n. 2, p. 661–686, 2012.

DETERDING, S.; DIXON, D.; KHALED, R.; NACKE, L. From game design elements to gamefulness: defining" gamification". In: *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments.* [S.l.: s.n.], 2011. p. 9–15.

DEURSEN, A. V.; ANICHE, M.; AUÉ, J.; SLAG, R.; JONG, M. D.; NEDERLOF, A.; BOUWERS, E. A collaborative approach to teaching software architecture. In: *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.* [S.l.: s.n.], 2017. p. 591–596.

FEICHAS, F. A.; SEABRA, R. D.; SOUZA, A. D. de. Gamificação no ensino superior em ciência da computação: Uma revisão sistemática da literatura. *Revista Novas Tecnologias na Educação*, v. 19, n. 1, p. 443–452, 2021.

FREEMAN, S.; EDDY, S. L.; MCDONOUGH, M.; SMITH, M. K.; OKOROAFOR, N.; JORDT, H.; WENDEROTH, M. P. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the national academy of sciences*, National Academy of Sciences, v. 111, n. 23, p. 8410–8415, 2014.

GALSTER, M.; ANGELOV, S. What makes teaching software architecture difficult? In: *Proceedings of the 38th International Conference on Software Engineering Companion.* [S.l.: s.n.], 2016. p. 356–359.

GAMMA, E. *Design patterns: elements of reusable object-oriented software.* [S.l.]: Addison-Wesley, 1995.

GAST, H. Patterns and traceability in teaching software architecture. In: *Proceedings of the 6th international symposium on Principles and practice of programming in Java.* [S.l.: s.n.], 2008. p. 23–31.

GEE, J. P. What video games have to teach us about learning and literacy. *Computers in entertainment (CIE)*, ACM New York, NY, USA, v. 1, n. 1, p. 20–20, 2003.

GIACOBO, D. Dbboard game: Um jogo de tabuleiro para o ensino e aprendizagem de conceitos de banco de dados. In: *Anais Estendidos do XXII Simpósio Brasileiro de Jogos e Entretenimento Digital.* Porto Alegre, RS, Brasil: SBC, 2023. p. 626–636. ISSN 0000-0000. Available from Internet: ⟨https://sol.sbc.org.br/index.php/sbgames_estendido/article/view/27854⟩.

GONÇALVES, A. C.; NETO, V. V. G.; FERREIRA, D. J.; SILVA, U. F. Flipped classroom applied to software architecture teaching. In: IEEE. *2020 IEEE Frontiers in Education Conference (FIE).* [S.l.], 2020. p. 1–8.

GRBAC, T. G.; CAR, Ž.; VUKOVIĆ, M. Requirements and architecture modeling in software engineering courses. In: *Proceedings of the 2015 European Conference on Software Architecture Workshops.* [S.l.: s.n.], 2015. p. 1–8.

HIDALGO, M.; ASTUDILLO, H.; CASTRO, L. M. How software architects learn: A pilot study of their learning style in kolb's learning styles inventory. In: IEEE. *2023 42nd IEEE International Conference of the Chilean Computer Science Society (SCCC).* [S.l.], 2023. p. 1–8.

KAZMAN, R.; KLEIN, M.; BARBACCI, M.; LONGSTAFF, T.; LIPSON, H.; CARRIERE, J. The architecture tradeoff analysis method. In: IEEE. *Proceedings. fourth ieee international conference on engineering of complex computer systems (cat. no. 98ex193)*. [S.l.], 1998. p. 68–78.

KEENAN, E.; STEELE, A. Exploring game architecture best-practices with classic space invaders. In: *Proceedings of the 1st International Workshop on Games and Software Engineering*. [S.l.: s.n.], 2011. p. 21–24.

KELLER, J. M. *Motivational design for learning and performance: The ARCS model approach*. [S.l.]: Springer Science & Business Media, 2009.

KIM, D.; KIM, S.; KIM, S.; PARK, S. Software engineering education toolkit for embedded software architecture design methodology using robotic systems. In: IEEE. *2008 15th Asia-Pacific Software Engineering Conference*. [S.l.], 2008. p. 317–324.

LAGO, P.; CAI, J. F.; BOER, R. C. de; KRUCHTEN, P.; VERDECCHIA, R. Decidarch: Playing cards as software architects. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HICSS). *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*. [S.l.], 2019. p. 7815–7824.

LAGO, P.; VLIET, H. V. Teaching a course on software architecture. In: IEEE. *18th Conference on Software Engineering Education & Training (CSEET'05)*. [S.l.], 2005. p. 35–42.

LELIS, M. R. L. Learn: Evolução de um jogo de tabuleiro para o ensino de arquitetura de software. 2024.

LIEH, O. E.; IRAWAN, Y. Exploring experiential learning model and risk management process for an undergraduate software architecture course. In: IEEE. *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.], 2018. p. 1–9.

LIEH, O. E.; IRAWAN, Y. Teaching adult learners on software architecture design skills. In: IEEE. *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.], 2018. p. 1–9.

LIMA, M. F.; MARQUES, A. B. dos S. Avaliação e melhoria da experiência do jogador em um jogo para ensino de arquitetura de software. In: SBC. *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*. [S.l.], 2024. p. 1442–1452.

MANNISTO, T.; SAVOLAINEN, J.; MYLLARNIEMI, V. Teaching software architecture design. In: *Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*. [S.l.: s.n.], 2008. p. 117–124.

MCGREGOR, J. D.; BACHMAN, F.; BASS, L.; BIANCO, P.; KLEIN, M. Using an architecture reasoning tool to teach software architecture. In: *20th Conference on Software Engineering Education Training (CSEET'07)*. [S.l.: s.n.], 2007. p. 275–282.

MENEZES, M. C. R. d.; VALLE, P. H. D.; OLIVEIRA, A. M. A card game for architectural patterns education. Submitted to the Brazilian Symposium on Computing Education (EduComp 2026), under review. 2026.

MENEZES, M. C. R. d.; VALLE, P. H. D.; OLIVEIRA, A. M. Two decades of software architecture education: State of the art, challenges, and future directions. In: *Proceedings of the 48th International Conference on Software Engineering*. Rio de Janeiro, Brazil: IEEE/ACM, 2026. (ICSE-SEET '26). Accepted for publication.

MONTENEGRO, C. H.; ASTUDILLO, H.; ÁLVAREZ, M. C. G. Atam-rpg: A role-playing game to teach architecture trade-off analysis method (atam). In: IEEE. *2017 XLIII Latin American Computer Conference (CLEI)*. [S.l.], 2017. p. 1–9.

MOURÃO, E.; PIMENTEL, J. F.; MURTA, L.; KALINOWSKI, M.; MENDES, E.; WOHLIN, C. On the performance of hybrid search strategies for systematic literature reviews in software engineering. *Information and software technology*, Elsevier, v. 123, p. 106294, 2020.

OLIVEIRA, B. R.; GARCÉS, L.; LYRA, K. T.; SANTOS, D. S.; ISOTANI, S.; NAKAGAWA, E. Y. An overview of software architecture education. In: SBC. *Congresso Ibero-Americano em Engenharia de Software (CIbSE)*. [S.l.], 2022. p. 76–90.

OUH, E. L.; GAN, B. K. S.; IRAWAN, Y. Did our course design on software architecture meet our student's learning expectations? In: IEEE. *2020 IEEE Frontiers in Education Conference (FIE)*. [S.l.], 2020. p. 1–9.

OUH, E. L.; IRAWAN, Y. Applying case-based learning for a postgraduate software architecture course. In: *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. [S.l.: s.n.], 2019. p. 457–463.

PETRI, G.; WANGENHEIM, C. Gresse von; BORGATTO, A. F. Meega+, systematic model to evaluate educational games. In: ______. *Encyclopedia of Computer Graphics and Games*. Cham: Springer International Publishing, 2017. p. 1–7. ISBN 978-3-319-08234-9. Available from Internet: ⟨https://doi.org/10.1007/978-3-319-08234-9_214-1⟩.

RAZMOV, V. Effective pedagogical principles and practices in teaching software engineering through projects. In: IEEE. *2007 37th Annual Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports*. [S.l.], 2007. p. S4E–21.

ROBIN, H.; MARC, L.; ROBERT, Z. A formal approach to game design and game research. *GDC. San Jose*, 2004.

RODRIGUES, C. S. C.; WERNER, C. M. Making the comprehension of software architecture attractive. In: IEEE. *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*. [S.l.], 2011. p. 416–420.

RUPAKHETI, C. R.; CHENOWETH, S. V. Teaching software architecture to undergraduate students: An experience report. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. [S.l.: s.n.], 2015. v. 2, p. 445–454.

RYAN, R. M.; KOESTNER, R.; DECI, E. L. Ego-involved persistence: When free-choice behavior is not intrinsically motivated. *Motivation and emotion*, Springer, v. 15, n. 3, p. 185–205, 1991.

SCANNAVINO, K. R. F.; NAKAGAWA, E. Y.; FABBRI, S. C. P. F.; FERRARI, F. C. Revisão sistemática da literatura em engenharia de software: teoria e prática. 2017.

SINGH, T. P.; RAO, T. Experiential learning: A systematic review of approach and learning models. *Library of Progress-Library Science, Information Technology & Computer*, v. 44, n. 3, 2024.

SOMMERVILLE, I. *Engenharia de Software (10ª edição)*. [S.l.]: São Paulo: Pearson, 2018.

SOUSA, T. A.; MARQUES, A. B. Learn board game: A game for teaching software architecture created through design science research. In: *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. [S.l.: s.n.], 2020. p. 834–843.

SOUZA, A. C. C.; SOUZA, F. C. M.; VILELA, R. F.; VALLE, P. H. Pmbok game ii: Um jogo educacional para apoiar o ensino de gestão de projetos de software. In: SBC. *Workshop sobre Educação em Computação (WEI)*. [S.l.], 2023. p. 454–464.

SOUZA, A. de; FREITAS, D. S.; RENGEL, D. M. Pedagogia sistêmica: experiência do unisenai campus joinville no ensino de arquitetura e desenvolvimento de software. In: SBC. *Escola Regional de Engenharia de Software (ERES)*. [S.l.], 2022. p. 181–189.

TEKINBAS, K. S.; ZIMMERMAN, E. *Rules of play: Game design fundamentals*. [S.l.]: MIT press, 2003.

TYREE, J.; AKERMAN, A. Architecture decisions: Demystifying architecture. *IEEE software*, IEEE, v. 22, n. 2, p. 19–27, 2005.

URREGO, J. S.; CORREAL, D. Archinotes: A tool for assisting software architecture courses. In: IEEE. *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*. [S.l.], 2013. p. 80–88.

VIDONI, M.; MONTAGNA, J. M.; VECCHIETTI, A. Project and team-based strategies for teaching software architecture. Tempus Publications, 2018.

WANG, A. I. Post-mortem analysis of student game projects in a software architecture course. In: IEEE. *2009 International IEEE Consumer Electronics Society's Games Innovations Conference*. [S.l.], 2009. p. 78–91.

WANG, A. I. Extensive evaluation of using a game project in a software architecture course. *ACM Transactions on Computing Education (TOCE)*, ACM New York, NY, USA, v. 11, n. 1, p. 1–28, 2011.

WANG, A. I.; ARISHOLM, E.; JACCHERI, L. Educational approach to an experiment in a software architecture course. In: IEEE. *20th Conference on Software Engineering Education & Training (CSEET'07)*. [S.l.], 2007. p. 291–300.

WANG, A. I.; STALHANE, T. Using post mortem analysis to evaluate software architecture student projects. In: *18th Conference on Software Engineering Education Training (CSEET'05)*. [S.l.: s.n.], 2005. p. 43–50.

WANG, A. I.; WU, B. Using game development to teach software architecture. *International Journal of Computer Games Technology*, Wiley Online Library, v. 2011, n. 1, p. 920873, 2011.

WEDEMANN, G. Scrum as a method of teaching software architecture. In: *Proceedings of the 3rd European Conference of Software engineering Education*. [S.l.: s.n.], 2018. p. 108–112.

WEI, B.; LI, Y.; DENG, L.; VISALLI, N. Teaching distributed software architecture by building an industrial level e-commerce application. *Software Engineering Research, Management and Applications*, Springer, p. 43–54, 2020.

WOUTERS, P.; NIMWEGEN, C. V.; OOSTENDORP, H. V.; SPEK, E. D. V. D. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of educational psychology*, American Psychological Association, v. 105, n. 2, p. 249, 2013.

WU, B.; STROM, J.-E.; WANG, A. I.; KVAMME, T. B. Xquest used in software architecture education. In: *2009 International IEEE Consumer Electronics Society's Games Innovations Conference*. [S.l.: s.n.], 2009. p. 70–77.

WU, B.; WANG, A. I. Comparison of learning software architecture by developing social applications versus games on the android platform. *International Journal of Computer Games Technology*, Wiley Online Library, v. 2012, n. 1, p. 494232, 2012.

WU, B.; WANG, A. I.; RUUD, A. H.; ZHANG, W. Z. Extending google android's application as an educational tool. In: IEEE. *2010 Third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning*. [S.l.], 2010. p. 23–30.

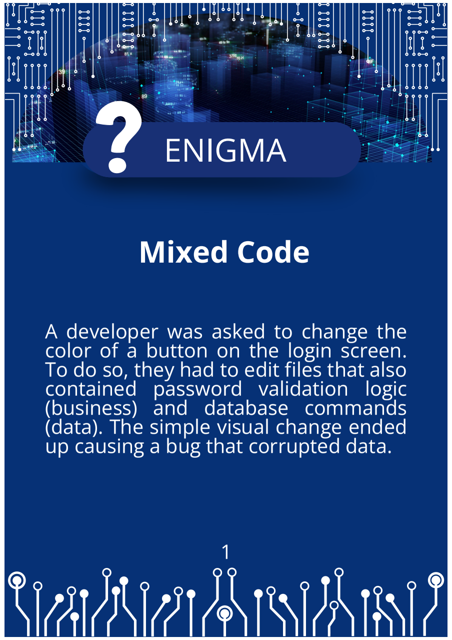
XAVIER, J.; WERNER, C.; TRAVASSOS, G. Uma abordagem para a seleção de padrões arquiteturais baseada em características de qualidade. In: *Anais do XVI Simpósio Brasileiro de Engenharia de Software*. Porto Alegre, RS, Brasil: SBC, 2002. p. 52–67. ISSN 2833-0633. Available from Internet: ⟨https://sol.sbc.org.br/index.php/sbes/article/view/23938⟩.

YÉPEZ, W. L. P.; ALEGRÍA, J. A. H.; BANDI, A.; KIWELEKAR, A. W. Training software architects suiting software industry needs: A literature review. *Education and Information Technologies*, Springer, v. 29, n. 9, p. 10931–10994, 2024.

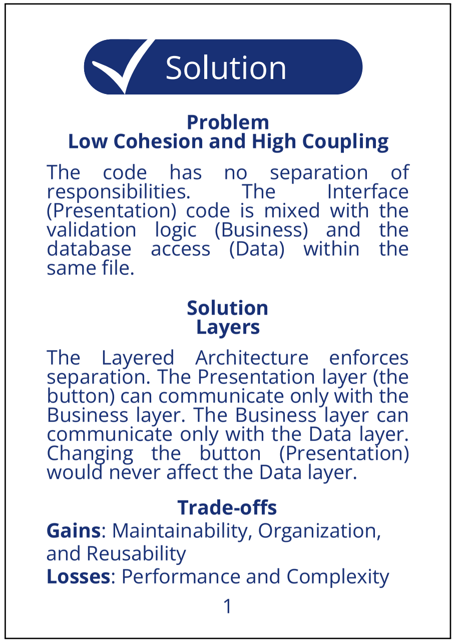
Appendices

A Game Cards

This appendix presents the complete collection of the 20 Mystery Cards developed for the Architectural Stories game. As detailed in Chapter 4, these cards serve as the core pedagogical artifact for fostering analytical reasoning and architectural diagnosis among students. Each card is displayed with its two complementary sides: the Enigma, which presents a narrative-driven scenario of a software system failure, and the Solution, which outlines the architectural root cause and recommends design patterns. The full set of visual designs and technical content for cards 1 through 20 is organized sequentially in Figures A.1 to A.5, illustrating the asymmetric information structure essential for the game's investigative dynamics.

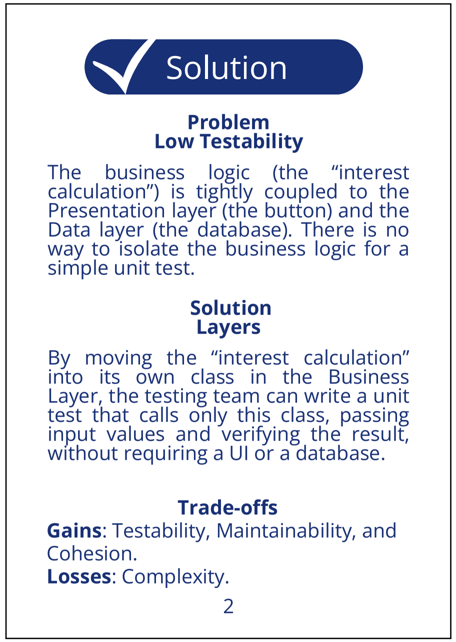


(a) Enigma 1

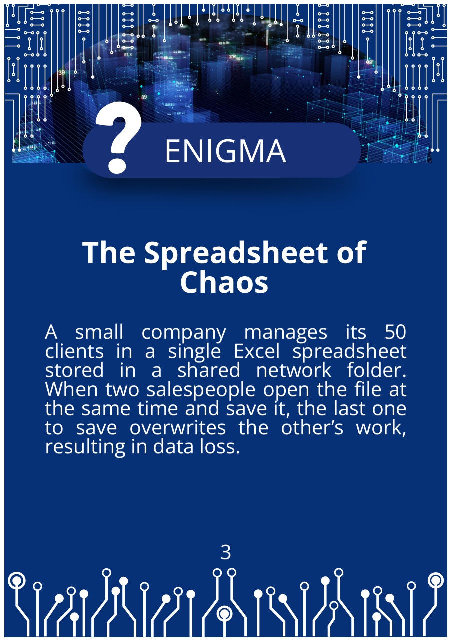


(b) Solution 1

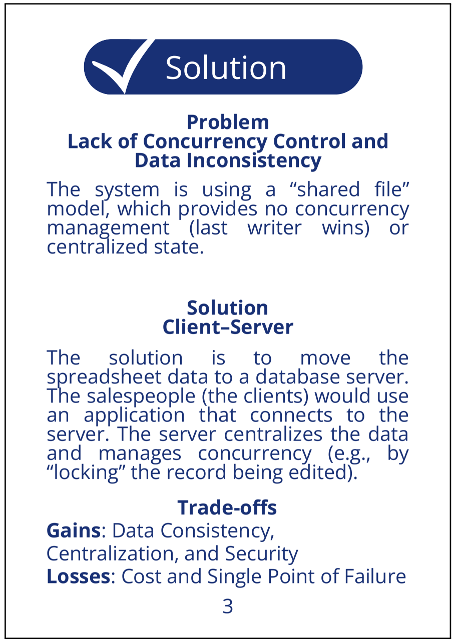
(c) Enigma 2



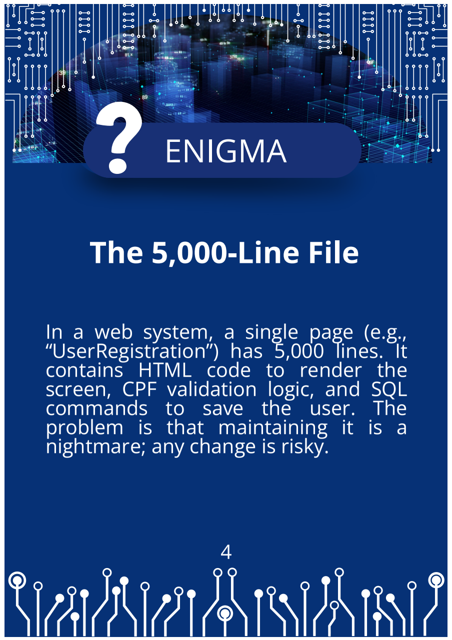
(d) Solution 2



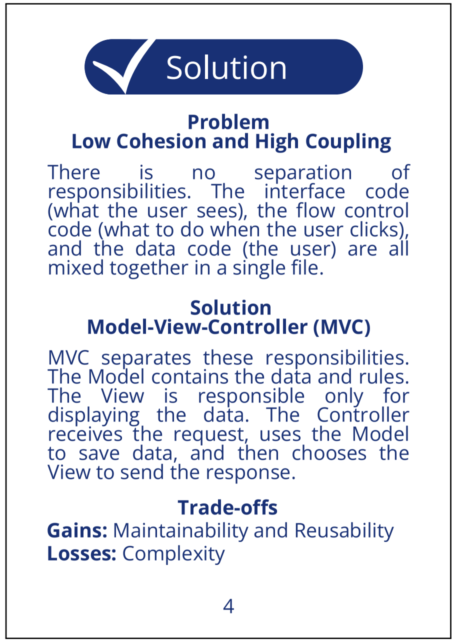
(e) Enigma 3



(f) Solution 3

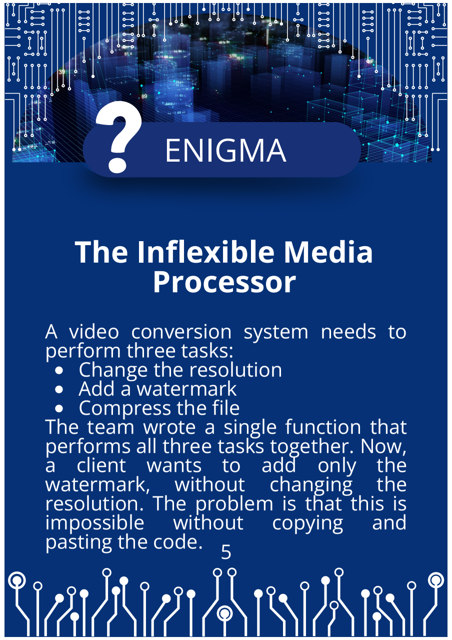


(g) Enigma 4

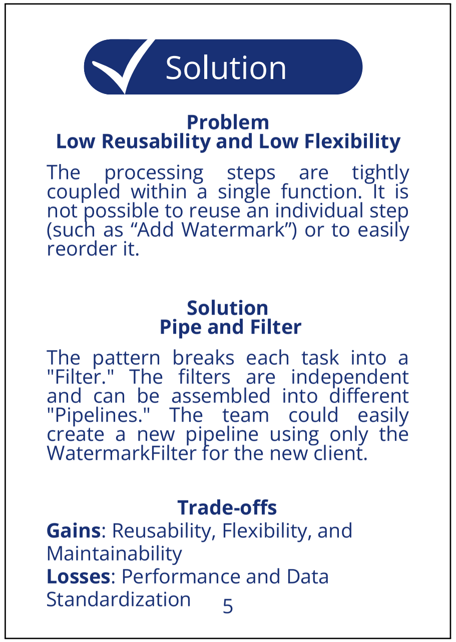


(h) Solution 4

Figure A.1: Game cards 1 to 4.

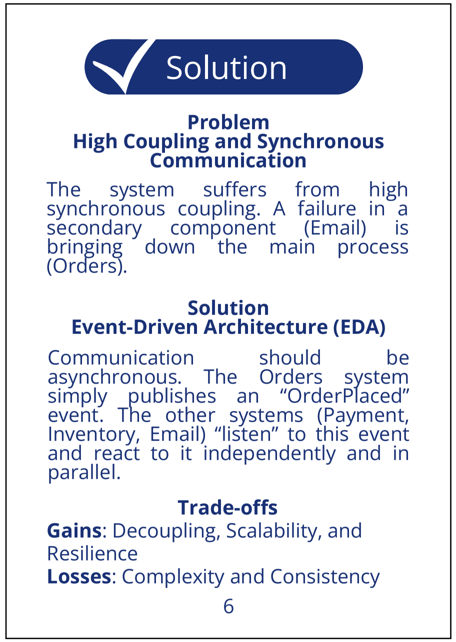


(a) Enigma 5

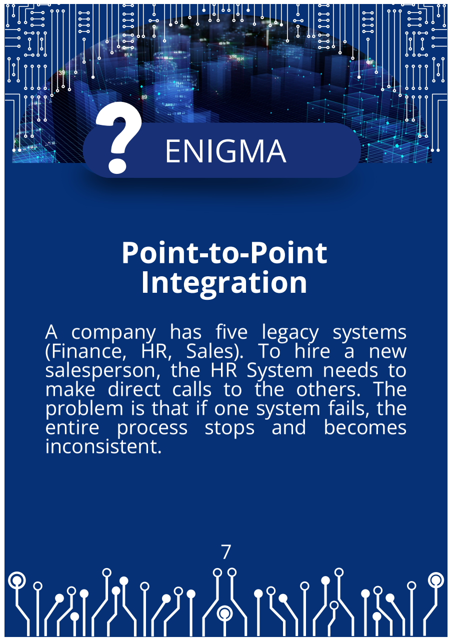


(b) Solution 5

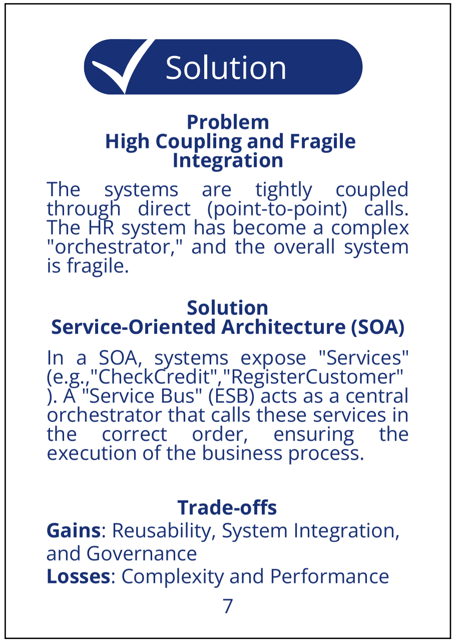
(c) Enigma 6



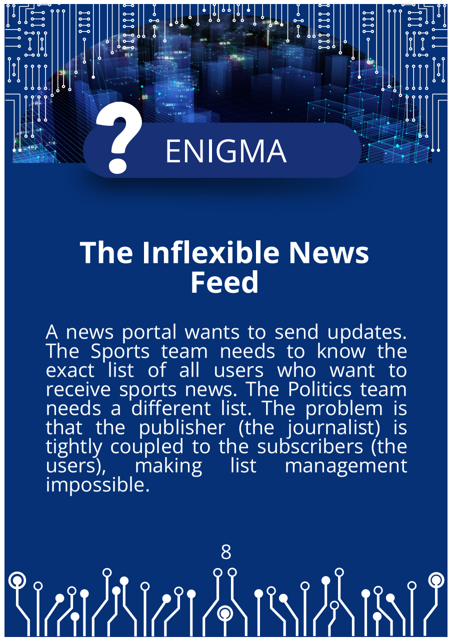
(d) Solution 6



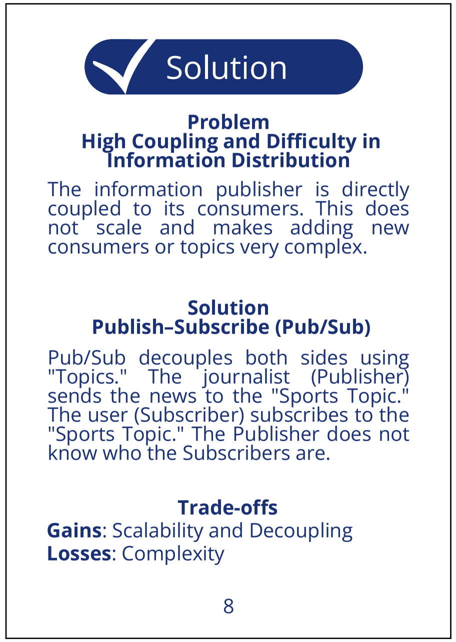
(e) Enigma 7



(f) Solution 7

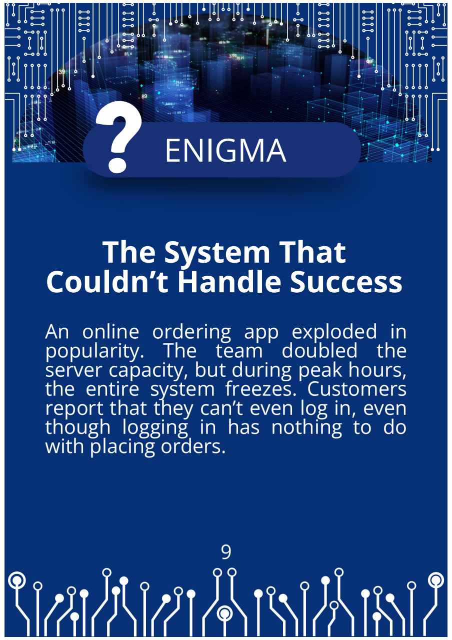


(g) Enigma 8



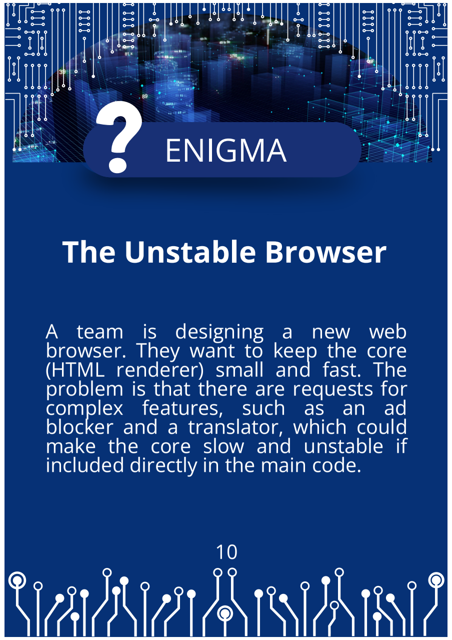
(h) Solution 8

Figure A.2: Game cards 5 to 8.

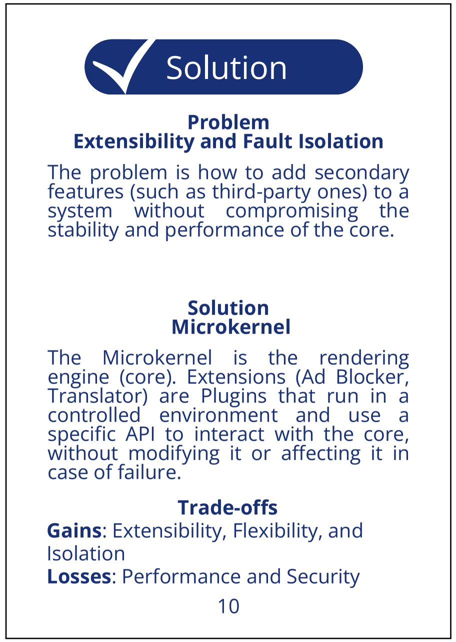


(a) Enigma 9

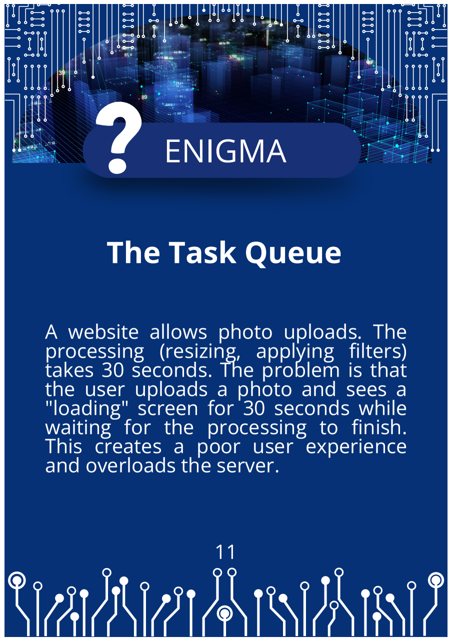
(b) Solution 9



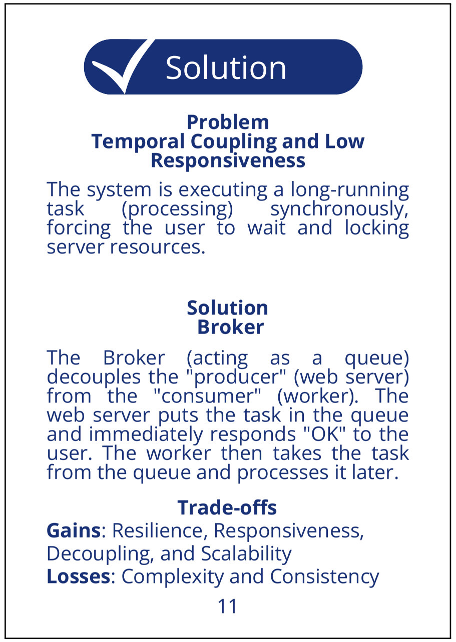
(c) Enigma 10



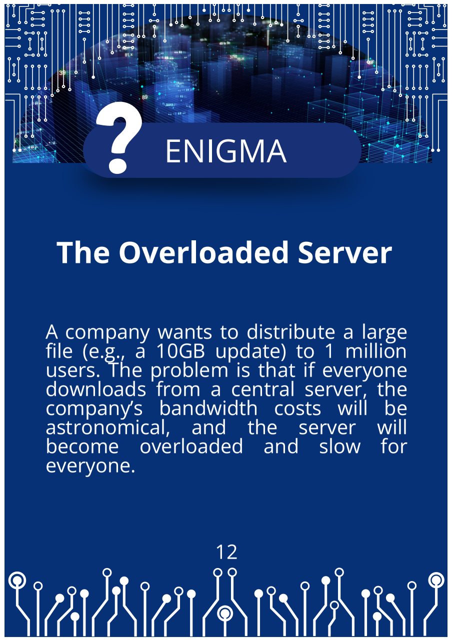
(d) Solution 10



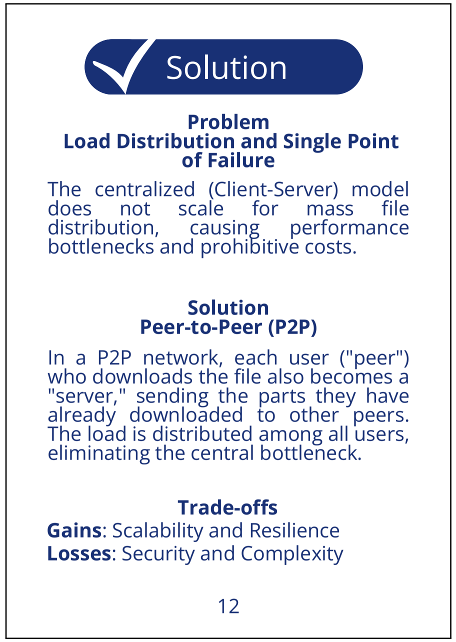
(e) Enigma 11



(f) Solution 11

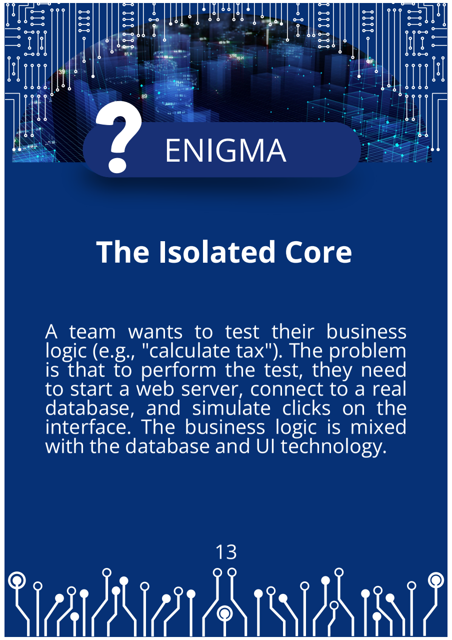


(g) Enigma 12

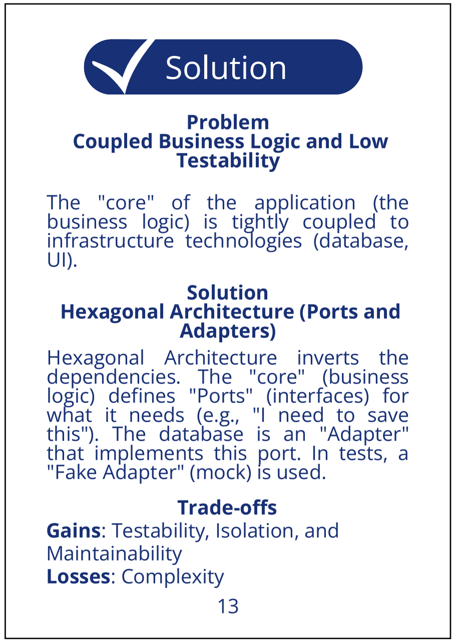


(h) Solution 12

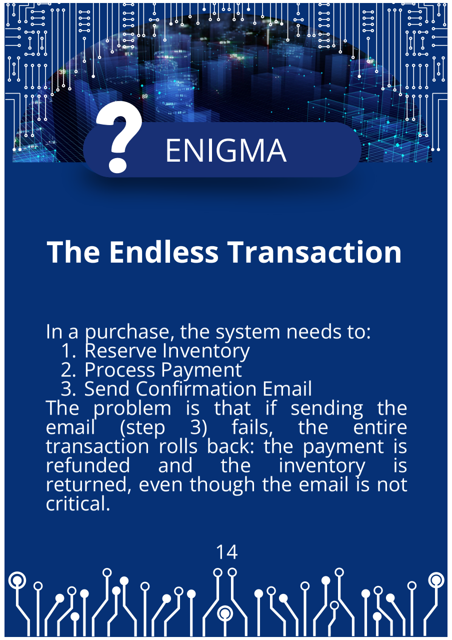
Figure A.3: Game cards 9 to 12.



(a) Enigma 13



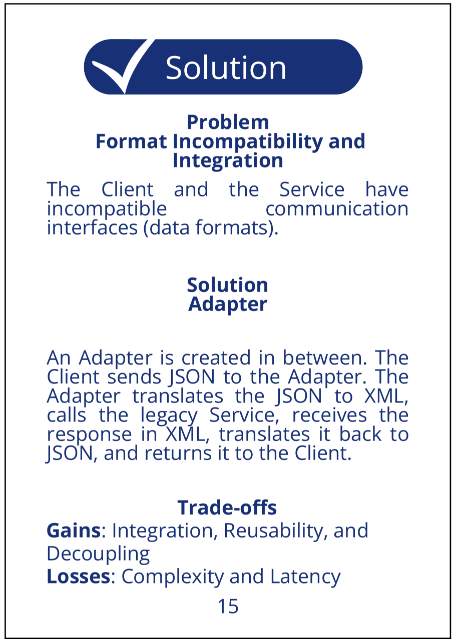
(b) Solution 13



(c) Enigma 14

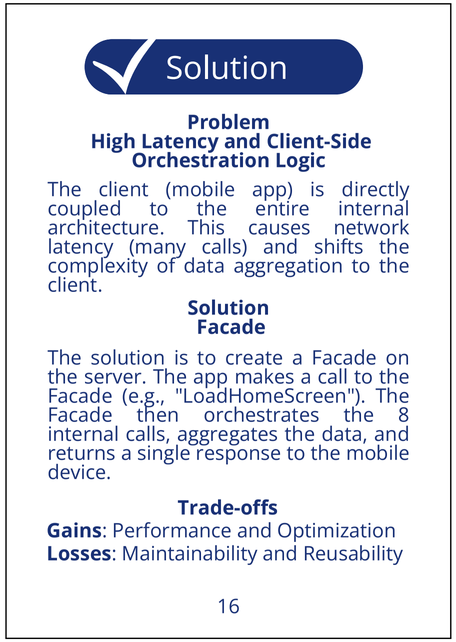
(d) Solution 14

(e) Enigma 15



(f) Solution 15

(g) Enigma 16



(h) Solution 16

Figure A.4: Game cards 13 to 16.

Figure A.5: Game cards 17 to 20.

B Manual of Architectural Stories

This appendix presents the full Instruction Manual for Architectural Stories, providing the operational framework necessary to implement the game in an educational or professional setting. As outlined in Section 4.4, the manual was designed to ensure that the game can be played autonomously by students, serving as the primary source for understanding roles, mechanics, and victory conditions. It details the specific responsibilities of the Architecture Master and the Investigators, the iterative process of inquiry through closed-ended questions, and the complete solution rule, which is central to the game's pedagogical goal of connecting symptoms to architectural patterns. By formalizing these procedures, the manual ensures a consistent and structured experience that balances competitive engagement with collaborative architectural reasoning.

Overview and Objective

Architectural Stories is a deduction game for 3–5 players. Players take on the role of Investigators trying to solve an "architectural crime" (a software problem) described on a card. To win the round, an Investigator must present the Complete Solution, which consists of both the Diagnosis (the problem) and the Prescription (the Architectural Pattern that solves it).

Player Roles

In each round, one player assumes the role of the Architecture Master (AM). The AM does not compete and is aware of the card's secret solution. Their sole responsibility is to judge the Accusations and answer the Investigators' questions.

The remaining players act as Investigators. Their objective is to ask questions and gather clues to decipher the architectural mystery presented in the round.

Game Flow

The game is divided into multiple rounds. At the beginning of each round, the AM draws a card and reads the Riddle (Side 1) aloud to all Investigators. Subsequently, the AM silently reads the Solution (Side 2) and keeps it secret.

The Investigators then begin the investigation by asking closed-ended questions to the AM. The AM may respond only with one of the following three words: “Yes”, “No”, or “Irrelevant”. Open-ended questions, such as “What is the problem?”, are not answered.

The round continues until an Investigator makes a Correct Accusation. At that point, the round ends and the AM reveals the complete solution aloud. For the next round, the AM role passes to the player on the left, who draws a new card and restarts the process.

The Complete Solution Rule (The Accusation)

At any moment during a round, an Investigator may make an Accusation. For the accusation to be considered Correct, it must include both:

- the Diagnosis (the architectural problem);
- the Prescription (the architectural pattern that solves it).

If the Investigator states only one of these elements, the Architecture Master must respond with “Incomplete”.

Scoring and Victory

The game uses an Insight Points (IP) board with a maximum of 8 points.

- If the Accusation is Correct, the Investigator gains +1 IP and the round ends;
- If the Accusation is Incomplete or Incorrect, the Investigator loses –1 IP.

After a correct accusation, the AM reads the complete solution aloud. The AM role then passes to the player on the left, and a new round begins.

The Investigator who reaches the end of the IP board, or who has the highest number of IP at the end of the game, is declared the winner.

Quick Guide for the Architecture Master

The role of the Architecture Master is to guide the players rather than act as an obstacle. The response “Irrelevant” should be used to redirect Investigators away from details that do not contribute to identifying the architectural solution, such as questions like “Was the system written in Python?”.

C Informed Consent Form

This appendix presents the Informed Consent Form (ICF) used in this study.

- **Voluntary Participation**: Your participation in this research is entirely voluntary and essential for obtaining meaningful insights that will help us better understand the quality of the game Architectural Stories;
- **Procedures**: Your participation in this study consists of answering a questionnaire containing both open-ended and closed-ended questions about the game;
- **Data Handling**: The information collected will be used exclusively for academic purposes. All data will be anonymized and processed securely, with full respect for the privacy of respondents;
- **Benefits**: You will not incur any costs or burdens by participating in this study, nor will you receive any type of reimbursement or compensation for authorizing the use of your data in the research;
- **Right to Refuse Participation**: You have the right to decline to participate in the research and may withdraw at any time, revoking your consent without any penalty, retaliation, or disadvantage.

Annexes

A Guidelines for Architectural Patterns in Architectural Stories

This appendix provides a comprehensive reference guide for the architectural patterns and principles integrated into the Architectural Stories game. As discussed in Section 4.4, these guidelines serve as the theoretical scaffolding designed to support players during the investigative process. Each entry summarizes the core intent, structural organization, and key trade-offs of a specific pattern. By consulting this material, players can move beyond intuitive guessing to evidence-based reasoning, allowing them to validate their hypotheses against established architectural theory and articulate their final solutions using precise technical vocabulary. This catalog ensures that the game functions not only as a diagnostic challenge but also as a continuous learning tool for Software Architecture.

Layers

Organizes the code into layers with specific responsibilities (e.g., Presentation, Business, Data), communicating only with the layer directly below. It provides high maintainability, organization, and separation of concerns. As drawbacks, it may introduce slight latency and increase initial complexity.

Client–Server

Divides the system into a Server (data and logic) and Clients (interface). It benefits from centralization, security, and ease of maintenance. Drawbacks include dependence on a single point of failure and the risk of server overload.

Model–View–Controller (MVC)

Separates the system into Model (data), View (interface), and Controller (control). It promotes separation of concerns, maintainability, and testability. The downside is increased complexity in managing state across components.

Pipe and Filter

Processes data in independent stages connected by "pipes," where the output of one filter becomes the input of the next. It offers flexibility, reusability, and testability, but may incur performance overhead and require standardized data formats.

Event-Driven Architecture

Components react to events (e.g., "OrderCreated") asynchronously, without direct dependency between producer and consumer. It provides decoupling, scalability, and resilience. Drawbacks include increased debugging complexity and eventual data consistency.

Service-Oriented Architecture (SOA)

Integrates enterprise systems through reusable business services, orchestrated by a Service Bus (ESB). It benefits from reuse, consistency, and centralized governance. Drawbacks include increased complexity and reduced team autonomy.

Publish–Subscribe (Pub/Sub)

Decouples publishers and subscribers through message topics. It offers high scalability and flexibility. Downsides include debugging difficulty and the lack of strong message delivery guarantees.

Microservices

Organizes the system into small, independent services, each with its own logic and database. It provides autonomy, scalability, and resilience. Trade-offs include operational complexity, latency, and debugging difficulty.

Microkernel

Defines a minimal core with essential functions, while additional functionality is provided through independent plugins. It offers extensibility, flexibility, and stability, but the API can be complex, and communication may be slow.

Broker

A messaging pattern in which an intermediary (broker) manages communication between producers and consumers (e.g., a task queue). It provides temporal decoupling, resilience, and load spike control. The main drawback is the risk of the broker becoming a single point of failure and a performance bottleneck.

Peer-to-Peer (P2P)

A decentralized network where there is no central server, and each node acts as both client and server simultaneously. The system becomes more scalable and resilient to failures, as each new node strengthens the network. However, it faces challenges related to security, efficient peer discovery, and the lack of centralized control.

Hexagonal Architecture (Ports and Adapters)

Separates the application core from the infrastructure through ports and adapters. It ensures high testability and flexibility, but has high initial complexity due to the large number of interfaces and layers.

Saga Pattern

Manages distributed transactions through local steps and compensating actions in case of failure. It maintains data consistency, but is very complex to implement and debug.

Adapter

A structural pattern that converts the interface of a component into another interface expected by the client, enabling integration between new and legacy systems. It promotes reuse and compatibility, but introduces an extra layer that can increase complexity.

Facade

Creates a simplified interface that encapsulates interaction with complex subsystems, hiding internal details and complexities. It eases usage, reduces coupling, and improves code readability, but may concentrate too much logic, becoming a critical maintenance point.

Proxy

An object that controls access to another object by "pretending" to be the real one. It is used to add functionalities such as security (permissions), caching (storing responses), or lazy loading. The benefit is separation of concerns; the drawback is slight latency and increased complexity.

Mediator

Centralizes communication among a set of objects ("colleagues"). Instead of communicating in an all-to-all manner, they interact only with the Mediator. The benefit is decoupling; the drawback is that the Mediator can become complex and a single point of failure.

Message Bus

A centralized communication infrastructure (a “bus”) where multiple systems (e.g., Finance, HR) connect to exchange messages. The benefit is standardized integration; the drawback is that the bus becomes a single point of failure and a complex bottleneck to manage.

Blackboard

Solves complex problems with no clear solution path (e.g., AI). Multiple independent “experts” collaborate by writing partial findings to a shared space (the “Blackboard”) until a solution emerges. The benefit is flexibility; the drawback is the extremely high complexity of implementation and debugging.

B Model for the Evaluation of Educational Games

This appendix presents the measurement instrument based on the MEEGA+, which was employed to evaluate the quality of Architectural Stories. The MEEGA+ is a specialized framework designed to assess educational games through two primary quality factors: Usability and Player Experience.

The complete set of items used in the evaluation is detailed in Table B.1, organized by their respective quality factors and dimensions, such as aesthetics, learnability, challenge, and social interaction. It is important to note that, for this study, items 10, 11, and 12 from the original MEEGA+ were excluded. This modification was necessary because those specific items address software customization and error protection, features exclusive to digital platforms that do not apply to a non-digital card game.

Table B.1: MEEGA+ items used in the evaluation.

Quality Factor	**Dimension**	**Item**
Usability	Aesthetics	1. The game's design is appealing (interface, graphics, board, cards, etc.).
Usability	Aesthetics	2. The text, colors, and fonts match and are consistent.
Usability	Learnability	3. I needed to learn only a few things before I could start playing the game.
Usability	Learnability	4. Learning to play this game was easy for me.
Usability	Learnability	5. I think most people would learn to play this game quickly.
Usability	Operability	6. I consider the game easy to play.
Usability	Operability	7. The game's rules are clear and understandable.
Usability	Accessibility	8. The fonts (size and style) used in the game are readable.
Usability	Accessibility	9. The colors used in the game are easy to understand.
Player Experience	Confidence	13. When I first looked at the game, I had the impression that it would be easy for me.
Player Experience	Confidence	14. The organization of the content helped me feel confident that I would learn from this game.
Player Experience	Challenge	15. This game is appropriately challenging for me.
Player Experience	Challenge	16. The game offers new challenges at an appropriate pace.
Player Experience	Challenge	17. The game does not become monotonous in its tasks.
Player Experience	Satisfaction	18. Completing the game's tasks gave me a sense of accomplishment.
Player Experience	Satisfaction	19. It is due to my personal effort that I am able to progress in the game.

Continuation of Table B.1

Quality Factor	Dimension	Item
Player Experience	Satisfaction	20. I feel satisfied with the things I learned in the game.
Player Experience	Satisfaction	21. I would recommend this game to my classmates.
Player Experience	Social Interaction	22. I was able to interact with other people during the game.
Player Experience	Social Interaction	23. The game promotes moments of cooperation and/or competition among players.
Player Experience	Social Interaction	24. I felt good interacting with other people during the game.
Player Experience	Fun	25. I had fun with the game.
Player Experience	Fun	26. Something happened during the game that made me smile.
Player Experience	Focused Attention	27. There was something interesting at the beginning of the game that captured my attention.
Player Experience	Focused Attention	28. I was so engaged in the game that I lost track of time.
Player Experience	Focused Attention	29. I forgot about the environment around me while playing this game.
Player Experience	Relevance	30. The game's content is relevant to my interests.
Player Experience	Relevance	31. It is clear to me how the game's content is related to the subject.
Player Experience	Relevance	32. The game is an appropriate teaching method for this subject.
Player Experience	Relevance	33. I prefer learning with this game rather than in another way.
Player Experience	Perceived Learning	34. The game contributed to my learning in the subject.
Player Experience	Perceived Learning	35. The game was effective for my learning compared to other activities.

C Intrinsic Motivation Inventory

This appendix presents the measurement instrument based on the IMI, which was used to assess the participants' subjective experience while playing Architectural Stories. The IMI is a multidimensional measurement device designed to evaluate intrinsic motivation and self-regulation through different psychological perspectives.

The specific version of the instrument selected for this study consists of 37 items, which are detailed and organized by their respective subscales in Table C.1. These items cover six dimensions: Interest/Enjoyment, Perceived Competence, Effort/Importance, Pressure/Tension, Perceived Choice, and Value/Usefulness. The items were evaluated using a 7-point Likert scale, ranging from ”not at all true” (1) to ”very true” (7).

As indicated in the table, items marked with an (R) are reverse-scored. This means that to calculate the final score, the participant's response is subtracted from 8 (e.g., $8 - \text{response}$). This procedure ensures that a higher final score consistently indicates a higher presence of the concept described by the subscale name (e.g., higher perceived competence or higher interest).

Table C.1: IMI items used in the evaluation.

Subscale	**Item**
Interest/Enjoyment	1. I enjoyed doing this activity very much.
Interest/Enjoyment	2. This activity was fun to do.
Interest/Enjoyment	3. I thought this was a boring activity. (R)
Interest/Enjoyment	4. This activity did not hold my attention at all. (R)
Interest/Enjoyment	5. I would describe this activity as very interesting.
Interest/Enjoyment	6. I thought this activity was quite enjoyable.
Interest/Enjoyment	7. While I was doing this activity, I was thinking about how much I enjoyed it.
Perceived Competence	8. I think I am pretty good at this activity.
Perceived Competence	9. I think I did pretty well at this activity, compared to other students.
Perceived Competence	10. After working at this activity for awhile, I felt pretty competent.
Perceived Competence	11. I am satisfied with my performance at this task.
Perceived Competence	12. I was pretty skilled at this activity.
Perceived Competence	13. This was an activity that I could not do very well. (R)
Effort/Importance	14. I put a lot of effort into this.
Effort/Importance	15. I did not try very hard to do well at this activity. (R)
Effort/Importance	16. I tried very hard on this activity.
Effort/Importance	17. It was important to me to do well at this task.

Continuation of Table C.1

Subscale	**Item**
Effort/Importance	18. I did not put much energy into this. (R)
Pressure/Tension	19. I did not feel nervous at all while doing this. (R)
Pressure/Tension	20. I felt very tense while doing this activity.
Pressure/Tension	21. I was very relaxed in doing these. (R)
Pressure/Tension	22. I was anxious while working on this task.
Pressure/Tension	23. I felt pressured while doing these.
Perceived Choice	24. I believe I had some choice about doing this activity.
Perceived Choice	25. I felt like it was not my own choice to do this task. (R)
Perceived Choice	26. I did not really have a choice about doing this task. (R)
Perceived Choice	27. I felt like I had to do this. (R)
Perceived Choice	28. I did this activity because I had no choice. (R)
Perceived Choice	29. I did this activity because I wanted to.
Perceived Choice	30. I did this activity because I had to. (R)
Value/Usefulness	31. I believe this activity could be of some value to me.
Value/Usefulness	32. I think that doing this activity is useful for improving my learning about the content.
Value/Usefulness	33. I think this is important to do because it can help me better understand the topic being studied.
Value/Usefulness	34. I would be willing to do this again because it has some value to me.
Value/Usefulness	35. I think doing this activity could help me to develop my problem-solving skills.
Value/Usefulness	36. I believe doing this activity could be beneficial to me.
Value/Usefulness	37. I think this is an important activity.

D Instructional Materials Motivation Survey

This appendix presents the measurement instrument based on the IMMS, used to evaluate the motivational aspects of the instructional material. The instrument is grounded in the ARCS model (Attention, Relevance, Confidence, and Satisfaction).

The instrument consists of 36 quantitative items distributed across four main categories, detailed in Table D.1. The items were evaluated using a 5-point Likert scale, ranging from "Not true" (1) to "Very true" (5).

Items marked with an (R) are reverse-scored. To calculate the final score for these items, the participant's response is subtracted from 6 (e.g., $6 -$ response). This ensures that higher scores consistently reflect higher levels of motivation across all dimensions.

Table D.1: IMMS items classified by ARCS dimensions.

Dimension	**Item**
Confidence	1. When I first looked at this lesson, I had the impression that it would be easy for me.
Attention	2. There was something interesting at the beginning of this lesson that got my attention.
Confidence	3. This material was more difficult to understand than I would like for it to be. (R)
Confidence	4. After reading the introductory information, I felt confident that I knew what I was supposed to learn from this lesson.
Satisfaction	5. Completing the exercises in this lesson gave me a satisfying feeling of accomplishment.
Relevance	6. It is clear to me how the content of this material is related to things I already know.
Confidence	7. Many of the pages had so much information that it was hard to pick out and remember the important points. (R)
Attention	8. These materials are eye-catching.
Relevance	9. There were stories, pictures, or examples that showed me how this material could be important to some people.
Relevance	10. Completing this lesson successfully was important to me.
Attention	11. The quality of the writing helped to hold my attention.
Attention	12. This lesson is so abstract that it was hard to keep my attention on it. (R)
Confidence	13. As I worked on this lesson, I was confident that I could learn the content.
Satisfaction	14. I enjoyed this lesson so much that I would like to know more about this topic.
Attention	15. The pages of this lesson look dry and unappealing. (R)
Relevance	16. The content of this material is relevant to my interests.
Attention	17. The way the information is arranged on the pages helped keep my attention.
Relevance	18. There are explanations or examples of how people use the knowledge in this lesson.
Confidence	19. The exercises in this lesson were too difficult. (R)
Attention	20. This lesson has things that stimulated my curiosity.
Satisfaction	21. I really enjoyed studying this lesson.
Attention	22. The amount of repetition in this lesson caused me to get bored sometimes. (R)

Continuation of Table D.1

Dimension	Item
Relevance	23. The content and style of writing in this lesson convey the impression that its content is worth knowing.
Attention	24. I learned some things that were surprising or unexpected.
Confidence	25. After working on this lesson for a while, I was confident that I would be able to pass a test on it.
Relevance	26. This lesson was not relevant to my needs because I already knew most of it. (R)
Satisfaction	27. The wording of feedback after the exercises, or of other comments in this lesson, helped me feel rewarded for my effort.
Attention	28. The variety of reading passages, exercises, illustrations, etc., helped keep my attention on the lesson.
Attention	29. The style of writing is boring. (R)
Relevance	30. I could relate the content of this lesson to things I have seen, done, or thought about in my own life.
Attention	31. There are so many words on each page that it is irritating. (R)
Satisfaction	32. It felt good to successfully complete this lesson.
Relevance	33. The content of this lesson will be useful to me.
Confidence	34. I could not really understand quite a bit of the material in this lesson. (R)
Confidence	35. The good organization of the content helped me be confident that I would learn this material.
Satisfaction	36. It was a pleasure to work on such a well-designed lesson.