Let's plAI: A Case Study on AI-Assisted Game Creation in Software Engineering Education

Rafael Parizi
Federal Institute Farroupilha
São Borja, Brazil
rafael.parizi@iffar.edu.br

Lorenzo Acosta
Federal Institute Farroupilha
São Borja, Brazil
lorenzo.68019@aluno.iffar.edu.br

Alexandre da Fonseca
Federal Institute Farroupilha
São Borja, Brazil
alexandre.2021307059@aluno.iffar.edu.br

Enzo Kleinpaul
Federal Institute Farroupilha
São Borja, Brazil
enzo.2021310072@aluno.iffar.edu.br

Abstract

This paper reports an exploratory case study in which 21 undergraduate students designed and built educational games about Software Engineering processes using Gemini Canvas and Generative AI over 16 class hours. Rather than consuming gamified content, students acted as game creators, translating theoretical concepts related to software process models and agile methods into interactive learning artifacts through iterative prompt engineering. We collected data through a structured questionnaire and prompt reports, analyzing them through descriptive statistics and STGT-informed qualitative coding. The qualitative analysis of prompt reports produced 37 open codes organized into five emergent categories: technical anchoring, engagement design, state control, narrative creation, and UX refinement. Students rated the support provided by Gemini Canvas for creating game logic at 4.43/5 and perceived content retention at 3.43/4, identifying quiz mechanics and immediate feedback as the most effective learning elements. All 21 participants recommended extending the approach to other courses. Results suggest that positioning students as active knowledge producers rather than passive consumers may support prompt engineering as an emergent competency and promote conceptual engagement with Software Engineering content, while also revealing context instability in GenAI interactions as a key practical limitation. Because the study did not include external performance measures, these findings should be interpreted as evidence of students' perceived learning and socio-technical interaction patterns rather than as measured learning gains.

Keywords

Software Engineering Education, Generative AI, Game Design

1 Introduction

Teaching Software Engineering (SE) is challenging in early undergraduate courses because students must understand abstract concepts such as processes, roles, artifacts, and decision-making. Lecture-based teaching may make these topics seem disconnected from real practice, limiting engagement and learning [7, 10, 25]. Thus, SE education has increasingly adopted active learning strategies to promote meaningful learning and closer contact with professional practices [14, 29].

In this context, educational game design has been increasingly discussed as a promising strategy for SE education [5, 13, 17]. By engaging students in the design of game mechanics, such as scoring systems, challenges, rules, and feedback, this approach moves learning beyond content reception and supports the active reconstruction of SE concepts. Prior studies suggest that game-based and gamified approaches may improve engagement, knowledge retention, and problem-solving [1, 4, 12]. In parallel, Generative Artificial Intelligence (GenAI) tools have expanded opportunities for technology-mediated knowledge construction, content generation, and rapid prototyping [16, 36]. When embedded in creative learning activities, GenAI may help students externalize and communicate their understanding through concrete artifacts. However, the combined use of educational game design and GenAI remains underexplored in SE education research.

This paper addresses this intersection by examining how educational game design and GenAI can be integrated in SE education. We do not treat this integration as a mere overlap of instructional resources. Instead, we frame game design as the pedagogical structure that organizes the learning activity, its objectives, and its artifacts, while GenAI serves as a computational environment that supports content generation, rapid prototyping, and iterative refinement. However, this integration also introduces challenges. Students must formulate effective prompts, critically assess AI-generated outputs, and ensure that the resulting artifacts remain conceptually aligned with SE concepts. Therefore, we investigate the need of pedagogical mediation to guide students' interaction with GenAI and reduce the risk of superficial or inaccurate learning outcomes [16].

We conducted an exploratory case study in which students designed and developed educational games about software processes using Gemini Canvas. We focused on software processes because they are among the first core concepts introduced in SE courses and often require students to understand roles, activities, artifacts, and workflows. Our study addresses the following research question: ***"How do students experience the combined use of GenAI and game design when creating educational games about SE processes?"***. We adopted *Gemini Canvas*[1] once it supports co-creation, iterative refinement, and interactive prototyping through successive interaction cycles. Our goal is not to compare different GenAI tools, but to examine how the integration of educational game design and GenAI can support students in constructing learning

[1] gemini.google.com

artifacts in SE education. This paper makes two main contributions: (i) a pedagogical approach that integrates educational game design and GenAI, positioning students as producers of learning artifacts rather than passive consumers of gamified content; and (ii) exploratory empirical evidence on students' perceived engagement, perceived comprehension of SE concepts, and socio-technical interaction strategies during AI-supported game creation.

The remainder of this paper is organized as follows: Section 2 presents the theoretical background; Section 3 describes our study, context, and data analysis procedures; Section 4 reports the results; Section 5 discusses findings and implications for educators and practice; Section 6 outlines threats to validity; and Section 7 provides final considerations and future work.

2 Background

2.1 Game Design in Education

Game design in education refers to pedagogical approaches in which students create educational games as part of the learning process [24, 35]. Rather than consuming pre-defined gamified content, students design rules, challenges, scoring systems, rewards, and feedback mechanisms as resources for knowledge construction. In Software Engineering, this approach is particularly relevant because it can help students engage with abstract concepts through the creation of concrete artifacts. Designing a game about a software process, for example, requires students to understand, organize, and communicate concepts such as roles, activities, artifacts, workflows, and decision points in a structured manner [32].

By acting as game designers, students become producers of educational artifacts rather than passive learners. This role encourages experimentation, revision, and knowledge construction [5, 8, 30], while also supporting skills central to SE education, such as problem-solving, critical thinking, communication, and collaboration [8]. By transforming SE concepts into challenges, rules, and objectives, students must evaluate whether their game mechanics are coherent, understandable, and aligned with learning goals. Thus, game design can bridge theory and practice by turning abstract SE concepts into interactive learning artifacts [3, 6].

2.2 Active Learning

Active learning encompasses pedagogical approaches in which students assume a central role in the learning process. Rather than passively receiving information, students engage in activities that require them to think, create, solve problems, collaborate, and reflect on their actions [29]. In this sense, active learning shifts instruction from content transmission to student participation, emphasizing learning as a process of construction, application, and reflection.

Empirical evidence supports the effectiveness of active learning. Freeman et al. [7], through a large-scale meta-analysis, showed that active learning improves student performance in science, technology, engineering, and mathematics when compared to traditional lecture-based instruction. This evidence is particularly relevant to computing education, since software development requires not only conceptual understanding but also practical, analytical, and collaborative skills that are difficult to develop through passive instructional formats alone.

Active learning also creates environments in which experimentation, error, feedback, and iteration become natural components of the learning process. This perspective aligns academic training with professional software development practices, where knowledge is continuously refined through problem-solving and artifact construction [2]. Educational game design mediated by GenAI fits within this perspective because students actively design, test, and refine interactive artifacts. By doing so, they externalize their understanding, evaluate design alternatives, and progressively construct knowledge about SE concepts.

2.3 Integrating GenAI, Game Design, and Active Learning in SE Education

Generative AI has gained a spotlight position in SE education as a tool to support creation, prototyping, problem-solving, and iterative feedback [16, 27]. In active learning contexts, these tools can expand opportunities for experimentation by enabling students to generate code, simulate scenarios, refine ideas, and evaluate outputs during the learning process [28]. A common element of this interaction is prompt engineering, which involves drafting and refining instructions to guide AI systems toward a specific goal [33, 34]. This practice requires students to organize their reasoning, clarify contexts, and critically validate AI-generated outputs, fostering analytical thinking and more conscious engagement with computational artifacts [9, 18].

Educational game design, GenAI, and active learning converge around student agency in knowledge construction. Game design structures learning through rules, challenges, goals, and feedback mechanisms, requiring students to transform SE concepts into coherent game dynamics [5, 12]. GenAI mediates the creative process by supporting experimentation, rapid prototyping, and iterative refinement [16, 27]. Active learning provides the pedagogical foundation, ensuring that students engage in reflection, decision-making, and meaningful artifact construction [7, 29]. Together, these elements position students as producers of knowledge rather than consumers of instructional content, aligning with learning perspectives in which understanding emerges through the creation of and reflection on artifacts [15, 19, 20, 26].

Although prior studies have examined gamification mainly as a consumption-oriented strategy [1, 5] and GenAI as a prototyping aid [16, 27], empirical studies exploring the synergy among GenAI, game design, and active learning in SE education remain limited. This gap motivates our study. We challenged students to design educational games about SE processes using GenAI within the Gemini Canvas environment. By positioning students as game *designers* rather than players, we aimed to examine how they experienced the combined use of GenAI and game design, including their perceived engagement, perceived understanding of SE concepts, prompt formulation strategies, and challenges related to conceptual fidelity in AI-supported educational game creation.

3 Research Method

This study investigates GenAI-supported game design as a pedagogical approach in SE education, using Gemini Canvas as the development environment. The tool allows students to generate, edit, and iteratively refine game code while keeping the conversational context visible. Rather than comparing GenAI tools, our focus is on how students transform software process concepts into game mechanics and interactive learning artifacts.

The following research question guides this study: ***“How do students experience the combined use of GenAI and game design when creating educational games about SE processes?”***. To answer it, we designed an exploratory single-case study with a qualitative approach, following Yin [37]. The case comprises a pedagogical activity that integrates educational game design and GenAI, while the unit of analysis is a Software Engineering class at the Federal Institute Farroupilha. The following sub-questions guided the investigation:

- **SQP1:** How did building game mechanics support students’ understanding of SE process concepts?
- **SQP2:** What challenges did students face when interacting with Gemini Canvas during game creation?
- **SQP3:** How did students perceive the overall contribution of the approach to their learning experience?

We conducted the study following a case study protocol, considering the stages illustrated in Figure 1 and described below.

Stage 1 – Contextualization and Planning. In Stage 1, the activity involved third-semester undergraduate students enrolled in the Software Engineering course at the Federal Institute Farroupilha (IFFar) and comprised 16 class hours. The professor introduced the activity, explained its integration of educational game design and GenAI, and presented a structured briefing[2] containing guidelines, requirements, and evaluation criteria. Each group selected one software process (Scrum, XP, Crystal Clear, Kanban, Waterfall or Spiral) and used Gemini Canvas to develop an interactive educational game about the selected process.

Stage 2 – Development. In Stage 2, after selecting their software process, the groups dedicated approximately 6 class hours to development, distributed across two 3-hour sessions. Students designed and refined their games by creating rules, scoring systems, glossaries, and interactive elements such as quizzes or simulations. During this process, they also documented the evolution of their prompts in a dedicated report.

Stage 3 – Project Evaluation. In Stage 3, students presented their projects during a 4-hour session. Each group demonstrated its final game and explained the main decisions made during development. The professor observed the solutions, the alignment between game mechanics and SE concepts, and the strategies adopted by the groups throughout the activity.

Stage 4 – Presentation and Data Collection. In Stage 4, we collected students’ perceptions of the learning process, the challenges they faced, and the perceived contributions of the approach. This stage lasted 4 hours. We used two complementary data sources: a questionnaire via Google Forms[3], described in Table 1, and the prompt reports submitted by the groups. These reports documented how students formulated, revised, and refined their instructions to Gemini Canvas. We combined these sources to triangulate students’ perceptions with evidence from their AI-supported co-creation process.

[2] https://doi.org/10.5281/zenodo.21226119
[3] https://docs.google.com/forms

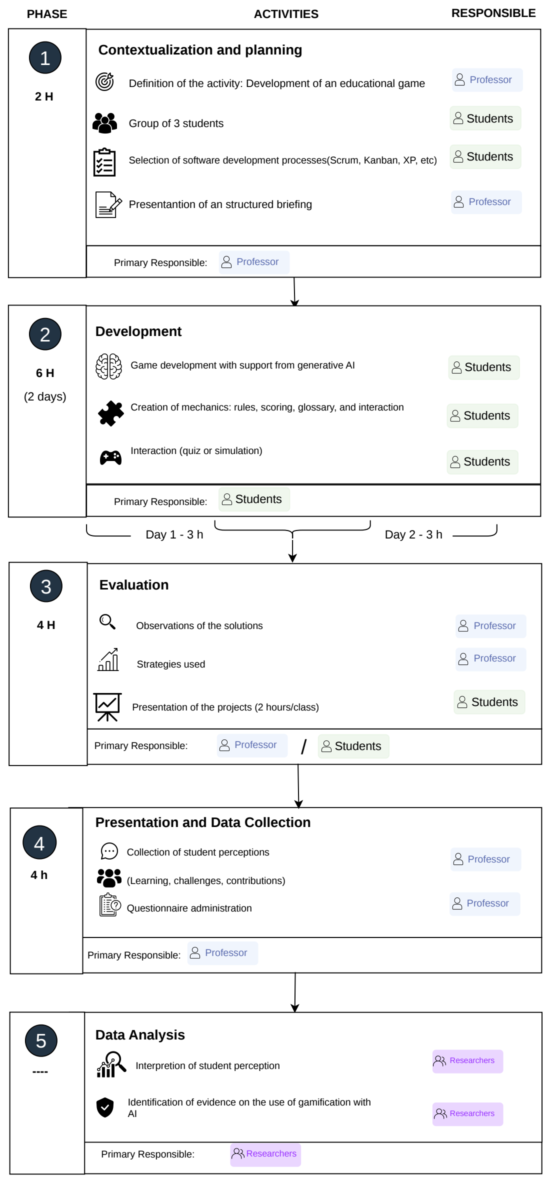


Figure 1: Research Method: Stages and Participants

Because the instructor also acted as one of the researchers, we adopted procedural precautions to reduce social desirability bias. The questionnaire was used only for research purposes and was not considered in students’ course grades. We anonymized the responses before analysis by replacing students’ identities with participant codes (P1–P21). We also informed students that critical comments about the activity and the tool were welcome and would not affect their evaluation in the course.

Table 1: Group Questionnaire: Game Design with GenAI

No.	Question	Response Type	SQP
Q1	Which software process did you choose for the game?	Open	–
Q2	Briefly explain the game and how it works.	Open	–
Q3	How much did the AI (Gemini) help in creating the game logic?	Likert (1–5)	SQP2
Q4	What was the biggest technical challenge during prompt evolution?	Open	SQP2
Q5	Describe a moment when you had to correct the AI.	Open	SQP2
Q6	Did creating rules and scoring improve your understanding?	Ordinal (1–4)	SQP1
Q7	Which game element best helped teach your peers?	Open	SQP1
Q8	How confident are you in explaining the process without support?	Ordinal (1–5)	SQP1
Q9	How did the AI change your perception of learning Software Engineering?	Open	SQP3
Q10	Should this method be used in other courses? Why?	Open	SQP3

Stage 5 – Data Analysis. In Stage 5, we analyzed the collected data using descriptive statistics and qualitative coding inspired by Socio-Technical Grounded Theory (STGT). We first summarized the closed-ended questionnaire responses and submitted artifacts according to the three research sub-questions.

For the qualitative analysis, two researchers used an independent-then-collaborative coding protocol. Researcher 1 performed line-by-line open coding of the prompt reports, with generative AI used only as auxiliary support for suggesting candidate labels. Researcher 1 reviewed and revised all suggested codes before retention. Researcher 2 then independently reviewed the code set, and disagreements were resolved through discussion until consensus. This process produced 37 open codes, later grouped into five emergent categories through axial coding. The same protocol was applied to the open-ended responses from Q5, Q9, and Q10.

Because the coding was applied to a finalized dataset, the study did not involve theoretical sampling, additional data collection guided by emerging categories, or substantive theory construction. Thus, we interpret the categories as exploratory socio-technical patterns, not as a full grounded theory. Similarly, questionnaire responses were treated as self-reported indicators of perceived learning, retention, and confidence, not as direct evidence of learning gains.

4 Results

4.1 Groups and Developed Games

The six groups independently selected different software process models as the conceptual basis for their games, without prior assignment by the professor. These choices covered both agile approaches, including Scrum, Kanban, XP, and Crystal Clear, and traditional models, including Spiral and Waterfall, as summarized in Table 2. Based on these selected processes, each group developed an educational game with specific mechanics, such as quizzes, challenges, simulations, scoring systems, and interactive decision-making elements. Figure 2 presents the interfaces of the games developed by the groups (in Portuguese because the study was conducted in an undergraduate class in Brazil)[4].

[4]https://zenodo.org/records/21226119

Questions Q1 and Q2 from the questionnaire (Table 1) provided complementary data about the software processes selected by the groups and the game mechanics of their projects.

Table 2: Software Processes Selected by Group

Group	Software Process	Game Genre	Members
G1	Spiral Model	Narrative / RPG	3
G2	Extreme Programming (XP)	Adventure / Racing	3
G3	Crystal Clear	Top-down / Action	3
G4	Scrum	Hangman / Puzzle	4
G5	Kanban	Simulation / Strategy	4
G6	Waterfall	Minigames / Sequential	4

G1 - Spiral Model. Corporate RPG where the student acts as a junior project manager. Errors in "knowledge battles" cost $50,000, and victory requires a 60% accuracy rate, simulating the cost of rework.

G2 - XP. Escape game where correct answers increase the distance from a giant boulder. Consecutive hit combos generate time bonuses, reinforcing the agile delivery pace under pressure.

G3 - Crystal Clear. Top-down game where the student manages a team's communication flow. Students must avoid "Noise Ghosts" that reduce visibility to 10%, simulating the impact of communicative noise.

G4 - Scrum. Themed hangman game where the student deciphers roles, artifacts, and ceremonies. Six errors trigger a mandatory review, reinforcing that learning in Scrum is continuous.

G5 - Kanban. Workflow simulation using drag-and-drop mechanics with Work-in-Progress limit of three simultaneous tasks. Exceeding the limit turns the column red and penalizes the team.

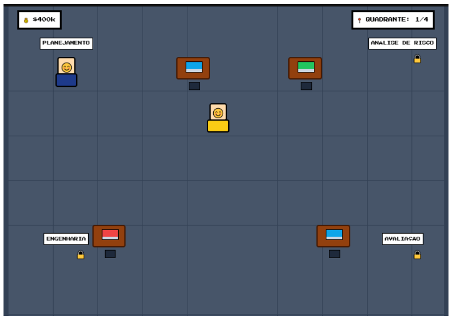
G6 - Waterfall. Five minigames represent the classical model phases. Failing any stage restarts the project from scratch; during Deploy, maintenance costs exceed the contract value.

4.2 Prompt Report Analysis

We analyzed the prompt reports to understand how students conducted co-creation with the AI, including emerging strategies, the navigation between disciplinary knowledge and game design, and model management throughout development. The analysis resulted in 37 open codes[5], which we grouped into five emergent categories with evidence extracted from the group responses.

Table 3 and Figure 3 were constructed through the STGT coding process described in Stage 5 (Section 3). The full chain of evidence is publicly available at the study's Zenodo repository.[6] We identified 37 open codes and organized them into five central categories that represent students' interaction strategies with Generative AI. The radial connections demonstrate how students anchor theoretical Software Engineering concepts (C1) and utilize gamification mechanics (C2) and narratives (C4) to materialize technical artifacts, while maintaining iterative control and usability refinement (C3 and C5) over the game's state. The distribution of codes, identified by group (G1–G6), highlights that technical anchoring and experience refinement were the most recurring learning dimensions during the pedagogical activity.

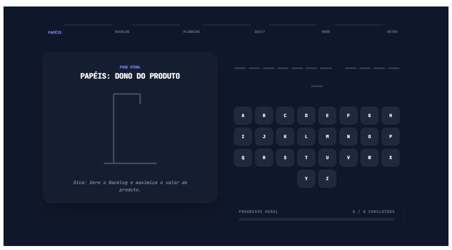
[5]Available at: https://zenodo.org/records/21226119
[6]https://doi.org/10.5281/zenodo.21226119



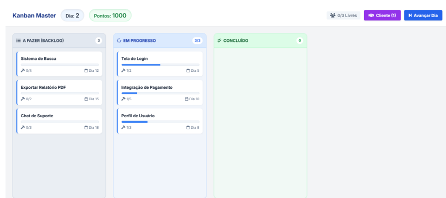
(a) G1 - Spiral

(b) G2 - XP

(c) G3 - Crystal Clear



(d) G4 - Scrum



(e) G5 - Kanban

(f) G6 - Waterfall

Figure 2: Educational game interfaces developed by the groups (in Portuguese).

Table 3: Emergent categories from the STGT-informed qualitative coding of student prompts

Category	Integrated Codes	Description	Evidence (excerpt)
C1 — Technical Anchoring and Conceptual Domain (Groups 1, 3, 4, 5)	- Software Thematic Conceptualization (G1) - Simulation Logic Specification (G4) - Tech Stack and Educational Domain Specification (G5) - Knowledge Domain Injection (G4) - Metaphorical Transposition of Agile Theory (G3) - Materialization of Technical Barriers (G3)	The student uses the prompt as a translation instrument: converting theoretical Software Engineering concepts (agile models, waterfall, kanban) into functional game specifications. The disciplinary domain structures the technical artifact from the beginning.	G3: *"Create a simple educational game inspired by the 'Osmotic Communication' principle of the Crystal Clear method..."* G4: *"Apply the concepts to the game: Product Owner, Scrum Master, Team, Product Backlog..."*
C2 — Gamification and Engagement Design (Groups 1, 2, 4, 5)	- Informal Punitive Gamification (G1) - Reward and Visualization Gamification (G4) - Performance Aesthetization (G2) - Positive Feedback Implementation (G2) - Punitive Logic and Engagement Refinement (G5) - Learning Progression Design (G4) - Rigid Progression Modeling (G1) - Methodological Constraint Coding (WIP) (G5)	Students build reward and punishment systems as a central learning mechanism. Points, streaks, penalties, and phase progression translate process rules into player behavior.	G2: *"...if the player gets 3 consecutive questions right, they gain 5 extra points, displaying the message: streak, +5 points"* G5: *"...deduct 100 points for each missed meeting... deduct 10 points for each overloaded day"*
C3 — State Control and Incremental Correction (Groups 1, 2, 3, 6)	- State Preservation and Control (G1) - Regression Management in Iterative Prompting (G3) - Logic Failure / State Bug Identification (G6) - Logic Inconsistency Correction (Bug Fixing) (G3) - Integrity Preservation Command (Lockdown) (G6) - Visual Feedback Simplification (G6) - Collision and Difficulty Calibration (G2)	An iterative vigilance pattern emerges: students monitor the artifact's state between interactions, identify regressions or logic bugs, and issue explicit restriction commands to the AI, revealing awareness of the risks of incremental development via prompts.	G6: *"DO NOT TOUCH ANY GAME, EMOJI, FUNCTIONALITY WORD, MINIGAME, NOTHING. ONLY change the name..."* G3: *"...restoration of the distraction box... which was accidentally removed in previous versions."*
C4 — Creation of Narratives and Humanization of the Artifact (Groups 1, 2, 5, 6)	- Emotional Feedback Personification (G1) - AI Politeness and Humanization (G1) - Narrative Urgency Metaphor (G2) - Narrative Contextualization (Onboarding) (G5) - Translation of Business Consequences into Narrative (G6) - Development Environment Visual Metaphor (G6) - Pedagogical Purpose Expansion (G2)	Students use narrative, emotion, and personification to make the game more immersive. This manifests in the artifact design (characters with emotional states, cultural metaphors) and in the relationship with the AI, treated as a sensitive social interlocutor.	G1: *"The game is too raw now, Gemini. Please, if it doesn't offend you, I would like you to bring more elements like Pokémon RPG, if you would be so kind..."* G6: *"...if the player has failed phases, a screen appears saying the system was completed but rejected by the client, with an invented sad character."*
C5 — Experience Refinement and Usability (Pedagogical UX) (Groups 2, 3, 4, 5, 6)	- Visual Ambience Refinement (G2) - Pedagogical Refinement (G1) - Technical Artifact Formalization (G2) - Usability Prioritization (UX) (G3) - Pedagogical User Experience Design (G4) - Autonomy and Educational Usability Design (G5) - Visual Obfuscation as Flow Challenge (G3) - Cognitive Interval Implementation (G3) - Aesthetic Smoothing of the Artifact (G3) - Rigorous Methodological Rule Flexibilization (G6) - Aesthetic-Functional Refinement (G4)	This is the broadest and most frequent category: it groups prompts that adjust the final user experience—visual clarity, learning flow, and cognitive load. The student acts as a pedagogical designer, balancing technical fidelity to the SE model with gameplay and accessibility.	G2: *"...fix the final scene which should display the cave entrance/exit with the boulder stopped... make the silhouette of a cave"* G5: *"After the game, it should present the main functions the user can perform within the interface"* G3: *"use graphic elements/icons instead of long texts."*

C1 - Technical Anchoring and Conceptual Domain (G1, G3, G4, G5). Six codes demonstrate that these groups converted theoretical SE concepts into functional specifications starting from the first prompt. G4 detailed game rules centered on agile roles:

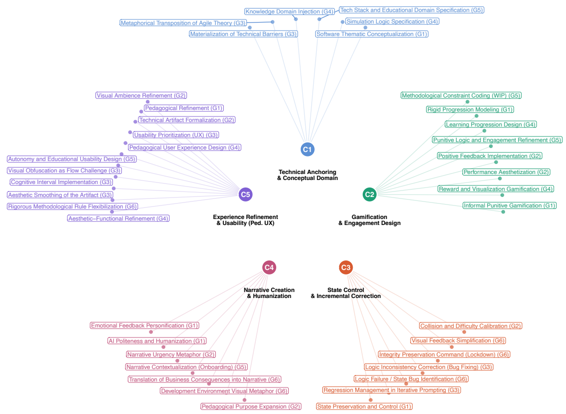


Figure 3: Radial distribution of open codes and emergent categories from the STGT-informed qualitative coding of student prompts.

"Apply the concepts to the game: Product Owner, Scrum Master, Team, Product Backlog...". G3 went further, transposing abstract principles into ludic mechanics: *"Create a simple educational game inspired by the 'Osmotic Communication' principle of the Crystal Clear method..."*. In these cases, students integrated learned concepts in the initial prompt, prioritizing theoretical grounding over gameplay.

C2 - Gamification and Engagement Design (G1, G2, G4, G5). Eight codes describe the construction of reward, punishment, and progression systems. G2 implemented positive reinforcement for streaks: *"...if the player gets 3 consecutive questions right, they gain 5 extra points, displaying the message: streak, +5 points"*. G5 applied penalties based on the professional impact of specific behaviors: *"...deduct 100 points for each missed meeting... deduct 10 points for each overloaded day"*. G1 adopted an informal and dramatic tone: *"...if he misses more than 2 questions, an image appears saying 'YOU'RE FIRED!'"*. SE concepts shifted from mere content to structures of incentive and consequence within the game.

C3 - State Control and Incremental Correction (G1, G2, G3, G6). Seven codes reveal a pattern of iterative vigilance: groups monitored the artifact between interactions, identified regressions, and issued explicit restrictions to the AI. G3 recovered functionalities lost between versions: *"...restoration of the distraction box... which was accidentally removed in previous versions"*. G6 resorted to a total lockdown: *"DO NOT TOUCH ANY GAME, EMOJI, FUNCTIONALITY WORD, MINIGAME, NOTHING. ONLY change the name..."*. This pattern indicates that, for these groups, each new prompt represented a potential risk to the progress made in previous rounds.

C4 - Creation of Narratives and Humanization of the Artifact (G1, G2, G5, G6). Seven codes demonstrate the use of narrative, emotion, and personification in both game design and the relationship with the AI. G1 treated the model as a social interlocutor: *"The game is too raw now, Gemini. Please, if it doesn't offend you, I would like you to bring more elements like Pokémon RPG, if you would be so kind..."*. G6 converted technical failure into a narrative element: *"...if the player has failed phases, a screen appears saying the system was completed but rejected by the client, with an invented sad character"*. The students perceived the tool not merely as a command executor, but as an agent capable of humanized communication.

C5 - Experience Refinement and Usability (Pedagogical UX) (G2, G3, G4, G5, G6). Eleven codes across five groups reveal student interest in visual clarity and learning flow, showing that participants progressively assumed the role of designers with pedagogical concerns. G3 prescribed visual preferences: *"use graphic elements/icons instead of long texts"*. G5 incorporated navigation guides into the game itself: *"After the game, it should present the main functions the user can perform within the interface"*. In all cases, the central focus shifted from technical functionality to the quality of the player's experience as a learner.

4.3 SQP1 — Perceived Content Retention and Student Autonomy

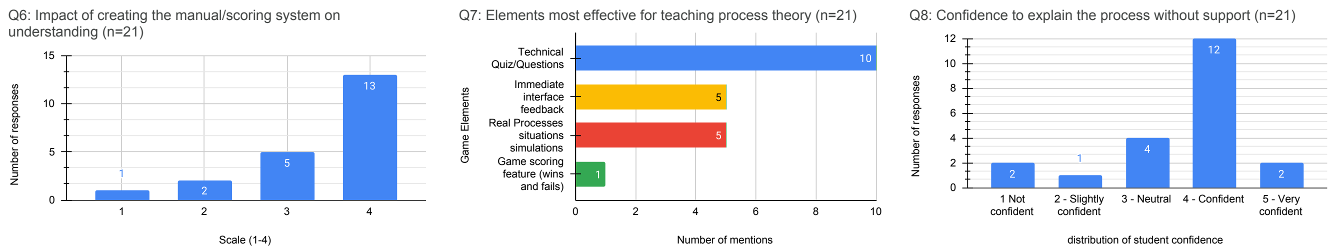
SQP1 investigated how the creation of mechanics contributed to the retention of SE concepts and student autonomy, based on Q6, Q7, and Q8 (Figure 4).

In Q6, the average score reached 3.43 (on a 1–4 scale), with 85.7% of students assigning a score of 3 or higher and 61.9% indicating the maximum level. Participants reported perceiving the activity as more effective for conceptual understanding than conventional reading. Regarding Q7, 47.6% of respondents identified the Quiz as the most effective mechanic for teaching content, followed by Real-World Situation Simulators and Immediate Feedback, both at 23.8%. This pattern aligns with category C2 of the prompt analysis: the reward and punishment systems constructed by the groups transform learning into a ludic context, which students commonly experience. In Q8, 66.7% of students declared themselves Confident or Very Confident in explaining their chosen process without support. However, crossing these results with Q6 reveals a relevant dissociation: P14, P19, and P21 perceived high content retention but low autonomy, indicating that internalizing content during creation does not guarantee the confidence to communicate it. Therefore, retention and autonomy represent related but partially independent dimensions in this activity.

4.4 SQP2 — Challenges in AI Interaction

We addressed SQP2 using questions Q3, Q4, and Q5. In Q3, the 21 participants rated how much Gemini Canvas facilitated the creation of game logic on a scale of 1 to 5. The perception was overwhelmingly positive: an average of 4.43 and a median of 5, with 57.1% assigning the maximum score and 28.6% assigning a 4. No participant recorded a score below 3. Figure 5 and Table 4 detail the distribution by group: G3 and G6 achieved a median of 5.0, while G2 showed the greatest internal variation (3.5). However, this positive outlook did not eliminate the interaction challenges evidenced in category C3 of the prompt analysis: even highly satisfied groups developed explicit state control and regression prevention strategies, indicating that the perceived ease of use still required active management of the AI's behavior throughout iterations.

Q4 asked participants to identify the greatest technical challenge in prompt evolution (Figure 6). The most frequent result was maintaining functional code logic between iterations, pointed out by 47.6% (10 of 21 students): when receiving new requests, the model frequently rewrote stable sections, forcing students into an active



(a) Content Retention (Q6). **(b) Most Effective Game Elements (Q7).** **(c) Level of Confidence (Q8).**

Figure 4: SQP1: Content retention, most effective mechanics, and autonomy perceived by students at the end of the activity.

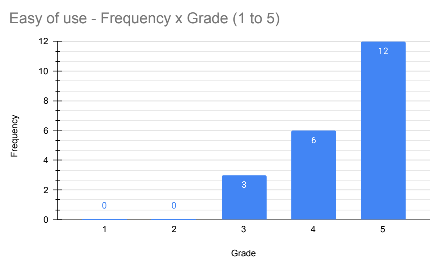


Figure 5: Distribution of Ease of Use – Q3 (n=21)

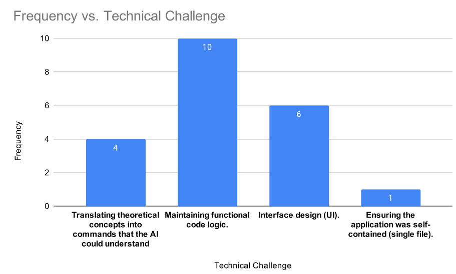


Figure 6: Distribution of challenges reported in Q4 (n=21)

Table 4: Q3: Ease of use of Gemini Canvas (n = 21).

Group	Participants (Rating)	Median	Description
G1	P1 (5), P2 (4), P3 (5), P4 (4)	4.5	Very positive
G2	P5 (4), P6 (3)	3.5	Positive with variation
G3	P7 (5), P8 (5), P9 (5)	5.0	Unanimously maximum
G4	P10 (5), P11 (4), P12 (5), P13 (3)	4.5	Very positive with variation
G5	P14 (4), P15 (5), P16 (4), P17 (5)	4.5	Very positive
G6	P18 (5), P19 (3), P20 (5), P21 (5)	5.0	Very positive with exception

Legend: 5 Facilitated greatly; 4 Facilitated; 3 Partially facilitated; 1|2 Did not facilitate (none)

role of verification and debugging. The codes in category C3 of the prompt analysis evidence this behavior, where groups even issued commands to block changes during the generation of new artifacts. Interface design was the second obstacle, with 28.6% of mentions: describing layouts via natural language required progressively more specific prompts. The translation of theoretical SE concepts into instructions understandable by the AI was cited by 19.0%, highlighting prompt engineering as a necessary competency for the effective pedagogical use of GenAI. Only 4.8% indicated the self-contained application restriction as a relevant barrier.

Question Q5 asked participants to describe a moment when they needed to correct the AI. The 21 responses underwent thematic analysis with open and axial coding [11], from which four categories and eight codes emerged (Table 5).

C1 — Instability of AI-generated context. Six participants reported situations where inserting new prompts triggered unsolicited changes in other parts of the project. The most frequent pattern (C1.1) was described by P1: *"When Gemini created something new, even when asked to keep everything the same, it ended up 'restarting' the project, keeping some random features of its choice and creating new ones, making the project take much longer"*. P20 and P21 reported the same behavior: *"When I asked it to change the home screen, it reinvented all the games"* and *"When I asked it to change the design of the home screen, it altered all the games"*. The insertion of external media (C1.2) was critical: P2 described that *"When we tried to insert a real image of a person, it led to a total code break, and we had to keep repeating the prompts"*, while P4 reported *"When asked to put the professor's photo in the avatar, the game changed completely"*. Finally, P11 (C1.3) noted: *"Sometimes I had to correct it because it did other things that strayed from what I wanted in the game"*.

C2 — Failure in process element representation. Six participants identified failures in the implementation of mechanics linked to technical content. P16 described (C2.1): *"The implementation of client meetings within the game was not satisfactory, requiring more than one prompt for the same functionality"*. P12 (C2.2) pointed out: *"It makes mistakes in things that are obvious to us, for example, giving space in the words of a word search"*. P18 summarized: *"Correcting a minigame that it did wrong"*. P3 (C2.3) focused the difficulty on the visual aspect: *"In the coding parts and especially in the design and character interaction part"*.

C3 — Iterative resolution via prompt reformulation. P16 and P17 described the strategy of adapting the prompt as a response to the AI's inadequate behavior. P17 synthesized: *"The meeting system was not implemented on the AI's first prompt attempt, but on the second attempt with an adapted prompt, the expected result was achieved"*.

Table 5: Categories and codes identified in the analysis of Q5 (n=21): AI correction moments during development

Category	Code	Description / Interpretation	Participants	Freq.
C1 — Instability of AI-generated context	C1.1 Project overwriting when editing isolated elements	When requesting a specific modification, the model rewrote previously stable sections, forcing students to review and reconstruct parts of the artifact.	P1, P19, P20, P21	4
	C1.2 Code collapse due to external media insertion	The inclusion of images or external files destabilized the generated code structure, causing errors that required manual intervention.	P2, P4	2
	C1.3 Behavior beyond requested scope	The model expanded the artifact in an unsolicited manner, introducing elements outside the scope defined by the group.	P11	1
C2 — Failure in process element representation	C2.1 Process mechanics not implemented on the first attempt	Translating software process rules into game logic required multiple iterations before producing the expected behavior.	P14, P16, P17, P18	4
	C2.2 Error in simple content elements	The model produced incorrect information regarding a directly verifiable technical concept, such as nomenclature or role definition.	P12	1
	C2.3 Difficulty in design and interaction of visual elements	Describing interface components through natural language resulted in layouts that diverged from the group's intent.	P3	1
C3 — Iterative resolution via prompt reformulation	C3.1 Adapted prompt as a deliberate correction strategy	Students recognized the failure, reformulated the prompt with greater specificity, and achieved the expected result in the subsequent interaction.	P16, P17	2
C4 — Absence of perceived intervention	C4.1 No necessary correction reported	Participants did not identify, in retrospect, any moments requiring explicit AI correction during development.	P5, P6, P7, P8, P9, P10, P13, P15	8

This behavior indicates that the student perceived the evolution of prompts as a necessity.

C4 — Absence of perceived intervention. Eight participants (38.1%) reported not having to correct the AI. P5 noted: *"Nothing of the sort happened"*; P15 stated: *"There was no moment where it made a mistake"*. These data may indicate that there are situations in which the AI is effective with low effort for generating games about specific content.

4.5 SQP3 — Impact on Learning and Approach Replicability

SQP3 investigated how students perceived the impact of using Gemini Canvas on their Software Engineering learning and their assessment of this approach's replicability in other disciplines. We analyzed the qualitative responses to questions Q9 and Q10.

Q9 requested that participants describe how Gemini Canvas changed their perception of learning SE. Five categories emerged from the thematic analysis (Table 6).

C1 — Transition from Theory to Practice.

P11 mentioned that *"the use of Gemini Canvas significantly changes the way of viewing SE learning because it shifts the focus away from just theory and leads directly to guided practice"*. P12 indicated that *"since SE is more abstract, the AI helped to materialize our ideas faster"*. P1 reported that the AI *"greatly facilitates time, and prototypes can be made faster; if someone doesn't know how to do it by hand, they will have an example to show their team"*. P15 wrote that *"the use of Gemini helped to better understand engineering concepts through the game, facilitating comprehension"*.

C2 – AI as a Content Mediator. P21 indicated that *"the AI itself helped me understand a lot about SE, explaining the waterfall model in a simplified way"*. P13 mentioned that the AI *"helped to understand better with fast and efficient answers and explanations"*. P3 stated that *"besides doing, it explains very well. I would use it always"*. P2 reported that using Gemini *"not only reinforced what I already understood about spiral but addressed more interesting details in a way that made learning through games more engaging"*.

Table 6: Categories and codes identified in the analysis of Q9 (n=21): Perception of learning SE with Gemini Canvas

Category	Code	Participants	Freq.
C1 — Transition from theory to practice mediated by AI	C1.1 Materialization of abstract concepts	P11, P12	2
	C1.2 Accelerated prototyping as a form of learning	P1, P15	2
C2 — AI as a mediator of content and explanation	C2.1 AI as a content explainer	P3, P13, P21	3
	C2.2 Reinforcement and deepening of prior knowledge	P2	1
C3 — Ludic and iterative learning as a break from the conventional	C3.1 Trial and error learning mediated by the game	P5, P6	2
	C3.2 Ludic and motivating character as a differentiator	P9, P14	2
	C3.3 Interactivity and visuality as engagement factors	P20	1
C4 — Surprise with the advancement and potential of AI	C4.1 Discovery of AI potential for system creation	P3, P19	2
	C4.2 Reflection on broader technological impact	P4	1
C5 — Absence of perceived impact or non-informative response	C5.1 Explicit absence of perceived change	P16, P17	2
	C5.2 Affirmative response without descriptive content	P7, P10, P18	3

C3 — Ludic and Iterative Learning. P5 mentioned that *"the game helps the player understand software processes simply through trial and error, along with the available theoretical material"*. P6 indicated that *"the trial and error game together with theoretical material helps a lot in learning about the processes"*. P14 reported that using AI *"showed me a fun way to learn the content, breaking away from the conventional"*. P9 also indicated that the AI approach was *"much more iterative and intuitive, sparking more interest"*. P20 added: *"A lot, as it is highly interactive and visual"*.

C4 — Surprise with AI Potential.

P19 wrote that they *"did not know that AI was so advanced at this point"*. P3 reported: *"we just kept giving the commands and it kept executing, we kept asking for adjustments and it kept doing them; it was very good to learn with it"*. P4 mentioned that it was *"quite surreal how easy it is to develop a system [...] it makes you think 'my God, industrial revolution 2.0'"*.

C5 — Absence of Impact or Non-Informative Response.

P16 and P17 denied any change (*"there was no perceived change"*; *"it didn't change"*.

Q10 asked if the methodology should be replicated in other disciplines. All 21 participants responded positively. The responses were organized into four categories (Table 7).

Table 7: Categories and codes identified in the analysis of Q10 (n=21): Replicability of the methodology in other disciplines

Category	Code	Participants	Freq.
C1 — Learning by creation as a pedagogical differentiator	C1.1 Game construction requires detailed research and study	P16, P17, P19	3
	C1.2 Transformation of complex content into something accessible	P1, P6, P15	3
	C1.3 Content retention as a result of creation	P13, P15	2
C2 — Engagement, creativity, and motivation	C2.1 Stimulation of curiosity and group work	P9, P13, P14	3
	C2.2 Alignment with the profile of the current generation	P5, P8	2
C3 — Conditional approval	C3.1 Replicability dependent on the discipline	P2, P7	2
	C3.2 Connection with content as a quality requirement	P20	1
	C3.3 Risk of AI limiting creative autonomy	P4	1
C4 — Concrete interest in specific disciplines	C4.1 Indication of discipline for replication	P12	1

C1 — Learning by Creation.

P16 mentioned that *"for the construction of the games, research on the chosen methodology was needed, which indeed helped in the construction of knowledge"*. P17 indicated that the AI *"encourages the detailed study of the content to create a game"*. P19 addressed that the AI *"encourages us to seek more of the content to be able to implement it in the game, thus learning the subject"*. P1 suggested that *"you use your mind to create your game while using items from the subject, thus assimilating something cool with the material that is usually tiring and repetitive"*. P6 indicated that it *"is a way to learn complex concepts in a more ludic and easier-to-digest way"*. P15 reported that the game *"will help to understand difficult concepts, improving memorization and study"*. Finally, P13 mentioned that the game *"stimulates imagination and creativity and even helps in content retention"*.

C2 — Engagement and Motivation.

P9 mentioned that the activity *"sparks curiosity and the group's willingness to think creatively without becoming an overwhelming process"*. P14 indicated that the activity was useful *"to explore students' creativity and potentially generate new ideas"*. P5 indicated that using GenAI as a learning strategy *"would make activities more intuitive and welcoming for us students, who are fully immersed in the daily use of AI"*. Finally, P8 highlighted that the approach could serve to *"update teaching methods in accordance with technology"*.

C3 — Conditional Approval.

P2 highlighted that GenAI with games might not always be applicable, indicating that *"there are subjects where I think it doesn't fit"*. Meanwhile, P7 commented that AI can help *"depending on the discipline, it could help a lot"*. P20 mentioned that if teachers *"can apply it in a way that is fully connected to the content, it would be a great activity"*. On the other hand, P4 signaled the risk of limiting creativity: *"yes, but without limiting the students' creativity and thinking only to what the AI provides"*.

C4 — Indication of Specific Disciplines.

P12 pointed out that the game + AI combination would help in other disciplines and expressed interest in trying it: *"I wanted to do it in Data Structures, for stacks, queues, etc."*, suggesting that the approach has potential for disciplines with a high degree of conceptual abstraction.

5 Discussion and Implications

The results suggest that positioning students as game creators, rather than game players, changes how GenAI is mobilized in the learning process. Prior work warns that LLMs may function as a *"crutch rather than a learning tool"* [21]. In our study, the production-oriented activity appeared to redirect this dynamic: students had to make conceptual decisions about software processes, rules, mechanics, feedback, and player interaction instead of simply asking Gemini Canvas for ready-made answers. This finding aligns with constructionist perspectives, in which learning emerges through artifact creation, and with Kasneci et al.'s argument that LLMs can support meaningful learning when they enable knowledge production rather than replace it [16].

However, the activity did not eliminate the tension between perceived learning and autonomous understanding. Some participants reported high content retention but low confidence in explaining the selected software process independently, which is consistent with the *"illusion of competence"* discussed by Mohamed et al. [23]. Moreover, students may respond differently to the same game design elements [22]. Therefore, content retention and communicable autonomy should be treated as related but distinct learning dimensions. Because we did not include external performance measures, we cannot claim demonstrated learning gains. The results should be interpreted as evidence of perceived conceptual engagement and socio-technical interaction patterns. Future replications should include individual, non-AI-mediated tasks and blind assessments of the games' conceptual fidelity by independent SE instructors.

A second theme concerns the pedagogical value of AI failures. When Gemini Canvas introduced regressions or misrepresented software process concepts, students had to revisit the original content, compare it with the artifact, and refine their prompts. This correction process mirrors Tolio et al., who found that 81.6% of Brazilian developers manually corrected errors after AI failures [31]. In our study, this competence emerged from the activity structure rather than explicit instruction, suggesting that creation-oriented tasks can turn the *"limitations and unexpected brittleness"* of LLMs [16] into productive friction.

A third observation concerns replicability and pedagogical design. Although all participants supported extending the approach to other courses, game mechanics must be structurally linked to theoretical content, not merely added as decorative elements. As Meißner et al. argue, educational game design must remain aligned with learning goals to avoid distraction [22]. Thus, the pedagogical value of GenAI depends less on the tool itself and more on the briefing, constraints, and assessment criteria that guide students' design decisions.

Implications for Education and Practice

For educators, the distinction between perceived retention and autonomous explanation suggests that self-reported instruments should be combined with unmediated performance tasks. Students should be asked to explain their artifacts, justify design decisions, identify conceptual inconsistencies, and transfer the embedded SE concepts without AI mediation.

For practice, providing access to GenAI is not sufficient. Instructors should design AI-supported creation tasks with explicit criteria for conceptual fidelity, not only technical functionality or visual quality. Correcting a programming bug and correcting a conceptual misrepresentation of Scrum, Waterfall, or another software process require different forms of reasoning; therefore, this distinction should be explicit in the activity design and assessment rubric.

6 Threats to Validity

Internal Validity: The study relied on students' self-perceptions and did not include a pre/post-test, a control group, a non-AI-mediated individual task, or a blind assessment of the games' conceptual fidelity. Thus, students may have overestimated their learning, and we cannot rule out an illusion of competence produced by rapid prototyping with GenAI. We mitigated this threat by triangulating closed-ended responses, open-ended explanations, and prompt reports; however, this supports claims about perceived learning and interaction strategies, not measured learning gains.

External Validity: The study involved 21 students from a single class, limiting generalization. To support transferability, we described the context, participants, activity, and use of Gemini Canvas in detail.

Construct Validity: The instruments measured perceived retention, usefulness, confidence, and autonomy rather than actual competence in Software Engineering process concepts. Therefore, we interpret the results as evidence of perceived learning support and conceptual engagement, not as objective evidence of learning improvement.

Conclusion Validity: The small sample limits statistical analysis and subgroup comparisons. Therefore, the findings are treated as exploratory, and we avoid causal or overly broad interpretations.

Social Desirability and Novelty Effects: The instructor also acted as researcher, which may have influenced students' responses, especially in a small class using a novel AI-based activity. We mitigated this threat by anonymizing responses, using participant codes, encouraging critical feedback, and triangulating questionnaires with prompt reports. Nevertheless, this risk cannot be fully eliminated, so we interpret positive responses cautiously.

Qualitative Analysis Scope: Although our coding was informed by Socio-Technical Grounded Theory, we did not conduct a full STGT study with theoretical sampling, iterative data collection, or substantive theory construction. The resulting categories should therefore be interpreted as exploratory socio-technical patterns from this case, not as a grounded theory of GenAI-supported game design in SE education.

7 Conclusion

This study investigated the creation of educational games supported by generative AI as a pedagogical strategy in Software Engineering education. Twenty-one students developed interactive games using Gemini Canvas, documenting the evolution of prompts and the decisions made throughout the process.

The results highlight three central findings. First, most students reported perceiving content retention, suggesting that the game creation process may have functioned as an active learning vector. Second, technical AI failures, rather than merely obstructing the work, induced involuntary conceptual revisions by forcing students to confront the original content to reformulate their prompts. Finally, the unanimous approval for replication in other disciplines indicates that the approach was perceived as pedagogically legitimate, not just as a technological novelty.

The results provide exploratory evidence that students perceived GenAI-supported game creation as an active learning experience. However, because we did not include external performance measures, we do not claim that the activity produced measurable learning gains. Instead, the study shows how students perceived the approach and how they used prompt engineering to translate Software Engineering process concepts into game mechanics.

The main contribution of this study is to demonstrate that, in creative tasks, the instability of LLMs can be converted into a pedagogical opportunity, provided the activity is structured around artifact creation rather than answer retrieval. The student ceases to be a consumer of answers and begins to act as a designer and evaluator of the generated artifact.

For future work, it would be interesting to expand the sample, include performance tests without AI assistance, and alternate AI-mediated activities with autonomous assessments to distinguish real learning from tool dependency.

Artifact Availability

The research artifacts produced in this study are publicly available on Zenodo. They can be accessed through the following DOI: https://doi.org/10.5281/zenodo.21226119.

Acknowledgements

Rafael Parizi acknowledges the support of FAPERGS under grant no. 25/2551-0000925-1. Enzo Kleinpaul also acknowledges FAPERGS for his scholarship under the same grant. Alexandre Pires and Lorenzo Campos acknowledge IFFar for their PIBIC-FAPERGS scholarships. Google Gemini Pro was employed to support the drafting of LaTeX tables and during the qualitative coding of student prompts.

References

[1] Manal M Alhammad and Ana M Moreno. 2018. Gamification in software engineering education: A systematic mapping. *Journal of Systems and Software* 141 (2018), 131–150.

[2] Tiago Silva Borges and Gidélia Alencar. 2014. Metodologias ativas na promoção da formação crítica do estudante: o uso das metodologias ativas como recurso didático na formação crítica do estudante do ensino superior. *Cairu em revista* 3, 4 (2014), 119–143.

[3] Patrick Buckley and Elaine Doyle. 2016. Gamification and student motivation. *Interactive learning environments* 24, 6 (2016), 1162–1175.

[4] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. 9–15.

[5] Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. 2015. Gamification in education: A systematic mapping study. *Journal of educational technology & society* 18, 3 (2015), 75.

[6] Adrián Domínguez, Joseba Saenz-de Navarrete, Luis De-Marcos, Luis Fernández-Sanz, Carmen Pagés, and José-Javier Martínez-Herráiz. 2013. Gamifying learning experiences: Practical implications and outcomes. *Computers & education* 63 (2013), 380–392.

[7] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the national academy of sciences* 111, 23 (2014), 8410–8415.

[8] Mourya Reddy Narasareddy Gari, Gursimran Singh Walia, and Alex David Radermacher. 2018. Gamification in computer science education: A systematic literature review. In *2018 ASEE Annual Conference & Exposition*.

[9] Juan Garzón, Eddy Patiño, and Camilo Marulanda. 2025. Systematic Review of Artificial Intelligence in Education: Trends, Benefits, and Challenges. *Multimodal Technologies and Interaction* 9, 8 (2025). doi:10.3390/mti9080084

[10] Carlo Ghezzi and Dino Mandrioli. 2005. The challenges of software engineering education. In *Proceedings of the 27th international conference on Software engineering*. 637–638.

[11] Hashini Gunatilake, John Grundy, Rashina Hoda, and Ingo Mueller. 2025. The Role of Empathy in Software Engineering - A Socio-Technical Grounded Theory. *ACM Trans. Softw. Eng. Methodol.* (Sept. 2025). doi:10.1145/3768315

[12] Juho Hamari, Jonna Koivisto, and Harri Sarsa. 2014. Does gamification work?– a literature review of empirical studies on gamification. In *2014 47th Hawaii international conference on system sciences*. Ieee, 3025–3034.

[13] Juho Hamari, Nannan Xi, Zampeta Legaki, and Benedikt Morschheuser. 2023. Gamification. In *Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences, 1105.

[14] Aoife Hennessy and Kieran Murphy. 2025. Barriers to student engagement: A focus group study on student engagement of first-year computing students immediate. *Irish Educational Studies* 44, 1 (2025), 183–200.

[15] Andri Ioannou and Brian E Gravel. 2024. Trends, tensions, and futures of maker education research: a 2025 vision for STEM+ disciplinary and transdisciplinary spaces for learning through making. *Educational technology research and development* 72, 1 (2024), 1–14.

[16] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences* 103 (2023), 102274.

[17] Georgios Lampropoulos, Euclid Keramopoulos, Konstantinos Diamantaras, and Georgios Evangelidis. 2023. Integrating augmented reality, gamification, and serious games in computer science education. *Education Sciences* 13, 6 (2023), 618.

[18] Daniel Lee and Edward Palmer. 2025. Prompt engineering in higher education: a systematic review to help inform curricula. *International Journal of Educational Technology in Higher Education* 22, 1 (2025), 7.

[19] Ilya Levin, Alexei Semenov, and Mikael Gorsky. 2025. Papert's Vision Realized: Constructionism and Generative AI. In *Constructionism Conference Proceedings*, Vol. 8. 419–426.

[20] Ilya Levin, Alexei L Semenov, and Mikael Gorsky. 2025. Smart learning in the 21st century: Advancing constructionism across three digital epochs. *Education Sciences* 15, 1 (2025), 45.

[21] May Mahmoud, Eric Asare, Nisa Shahid, Nourhan Sakr, and Sarah Nadi. 2026. Fast and Fleeting: Evaluating ChatGPT's Impact on Students' Computational Thinking Skills. In *Proceedings of the 2026 IEEE/ACM 19th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE/ACM, Rio de Janeiro, Brazil. CHASE 2026.

[22] Niklas Meißner, Lukas Trautwein, Sandro Speth, Nadine Nicole Koch, and Steffen Becker. 2026. Utilizing Adaptive Gamification in an Intelligent Tutoring System for Software Engineering Education. In *Proceedings of the 2026 IEEE/ACM 38th International Conference on Software Engineering Education and Training (ICSE-SEET)*. IEEE/ACM. ICSE-SEET 2026.

[23] Suad Mohamed, Abdullah Parvin, Najma Ismail, Kimberly Amaya, Esteban Parra, and Michael Oliver. 2026. Beyond the Prompt: How Novices Engage with AI During Code Summarization. In *Proceedings of the 2026 IEEE/ACM 38th International Conference on Software Engineering Education and Training (ICSE-SEET)*. IEEE/ACM. ICSE-SEET 2026.

[24] Pablo Moreno-Ger, Daniel Burgos, Iván Martínez-Ortiz, José Luis Sierra, and Baltasar Fernández-Manjón. 2008. Educational game design for online education. *Computers in Human Behavior* 24, 6 (2008), 2530–2540. doi:10.1016/j.chb.2008.03.012 Including the Special Issue: Electronic Games and Personalized eLearning Processes.

[25] Sofia Ouhbi and Nuno Pombo. 2020. Software engineering education: Challenges and perspectives. In *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 202–209.

[26] Seymour A Papert. 2020. *Mindstorms: Children, computers, and powerful ideas*. Basic books.

[27] Alexandre Pires, Enzo Kleinpaul, Lorenzo Campos, and Rafael Parizi. 2025. From Prompts to Prototypes: Exploring ChatGPT in Collaborative Software Prototyping Education. In *Simpósio Brasileiro de Qualidade de Software (SBQS)*. SBC, 453–462.

[28] Laszlo Pokorny. 2025. Gamified Artificial Intelligence in Education: A Systematic Literature Review. (11 2025). doi:10.5281/zenodo.17578775

[29] Michael Prince. 2004. Does active learning work? A review of the research. *Journal of engineering education* 93, 3 (2004), 223–231.

[30] Bruno Amarante Couto Rezende and Vânia dos Santos MESQUITA. 2017. O uso de gamificação no ensino: uma revisão sistemática da literatura. *SBC–Proceedings of SBGames* (2017), 1004–1007.

[31] Antônio Sergio Tolio, Enzo Becker, Felipe Bagesteiro, Cristiano Tolfo, and Lisandra Fontoura. 2026. LLMs in Practice: Results from a Survey with Software Development Professionals in Brazil. In *Anais do XXII Simpósio Brasileiro de Sistemas de Informação* (Vitória/ES). SBC, Porto Alegre, RS, Brasil, 636–653. doi:10.5753/sbsi.2026.248579

[32] Simone Tonhão, Marcelo Shigenaga, Julio Herculani, Andressa Medeiros, Aline Amaral, Williamson Silva, Thelma Colanzi, and Igor Steinmacher. 2023. Gamification in software engineering education: A tertiary study. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 358–367.

[33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[34] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).

[35] C. H. Wu, Y. C. Chien, M. T. Chou, et al. 2025. Integrating computational thinking, game design, and design thinking: a scoping review on trends, applications, and implications for education. *Humanities and Social Sciences Communications* 12, 163 (2025). doi:10.1057/s41599-025-04502-x

[36] Amir Yavariabdi, Bhuwan Paudel, Tamara Carleton, and Carlos Diego Andrade De Almeida. 2025. Generative AI in assessment and feedback generation in higher education: A systematic review. In *2025 International Conference on Education Technology and Computers (ICETC)*. IEEE, 361–371.

[37] Robert K. Yin. 2018. *Case Study Research and Applications: Design and Methods* (6th ed.). Sage Publications, Thousand Oaks, CA.