Informatics in Education, 2025, Vol. 24, No. 2, 223–260

DOI: 10.15388/infedu.2025.11

MPS Manager: A Serious Game to Enhance Teaching on Software Development Process Quality

Iago Rodrigues AGUALUZA[1], Silvana Morita MELO[2], Lina GARCÉS[3], Vânia de Oliveira NEVES[1,*]

[1]*Fluminense Federal University, Brazil*
[2]*Federal University of Grande Dourados, Brazil*
[3]*University of São Paulo, Brazil*
e-mail: iagoagualuza@gmail.com, silvanamelo@ufgd.edu.br, linagarces@usp.br, vania@ic.uff.br

Received: October 2024

Abstract. Modern software companies prioritize high-quality products for competitiveness, and Software Process Improvement (SPI) models help achieve this. In Brazil, the Brazilian Software Process Improvement Model (*MPS-SW* model) is widely used, but its complexity and extensive documentation make it challenging to teach in undergraduate courses. To address the lack of students engagement to learn SPI, we developed the *MPS Manager*, a serious game that incorporates gamification to facilitate learning about the *MPS-SW* model. The game was evaluated in four Software Engineering courses across three universities with 83 students. Using the Model for the Evaluation of Educational GAmes (*MEEGA+*) method, students assessed the game across dimensions such as usability, confidence, and learning, with 55% overall agreement. Further analysis explored correlations between satisfaction and factors like gender, gaming experience, and course format (i.e., virtual or in-person). Feedback from students highlighted the need for improved engagement, social interaction, and reduced gameplay monotony, which will guide future game enhancements.

Keywords: MPS-SW, software process quality, software engineering education, serious game, gamification.

1. Introduction

Teaching Software Engineering presents significant challenges, as it requires students to simultaneously develop technical and interpersonal skills in order to solve real-

* Corresponding author.

world problems through the development of high-quality software (Ghezzi and Mandrioli, 2005). One of the core topics addressed in these courses is software process improvement (SPI), whose understanding demands that students apply various software engineering techniques throughout the entire development life cycle, from process selection to product maintenance (Pinedo *et al.*, 2023). Mastering these concepts requires high levels of abstraction, enabling students to comprehend the process as a whole, the different categories of quality, and the technical details involved in each activity.

However, as pointed out by Shaw (2000), the Software Engineering curriculum has historically emphasized coding and debugging over early-stage activities such as analysis and specification. This curricular orientation may be reflected in students' behavior, as they often show a preference for programming activities – perceived as more practical and engaging – over abstract practices such as requirements elicitation and modelling. This behavior reflects a broader gap between theoretical learning in the classroom and real-world industry contexts, making it particularly challenging to teach process-oriented topics like SPI, especially in undergraduate programs where students often lack practical experience with real-world projects.

This difficulty becomes even more pronounced when teaching SPI models like Capability Maturity Model Integration (*CMMI*) and the Brazilian Software Process Improvement Model (*MPS-SW*), due to their abstract and complex nature (De Oliveira Colares *et al.*, 2023). These models present challenges for educators because of the lack of practical examples of their application in realistic projects, their extensive documentation, and the interdisciplinary knowledge required to understand and apply them in the classroom, making it difficult for students to grasp the utility of such quality models in their professional careers. In this context, students may have some misconceptions about the importance of learning this topic, mainly because they cannot visualize this knowledge as required for working as a software engineer (Gold-Veerkamp, 2021). As a result, studying these models is often perceived as tedious, prompting educators to adopt creative strategies to engage students and enhance their understanding.

Given these challenges, it becomes necessary to explore alternative instructional strategies capable of engaging students and improving their understanding of complex process-oriented models. In this context, serious games emerge as a potential solution to create interactive and simulated environments that make it easier to learn complex subjects, offering students a more hands-on experience (Ouhbi and Pombo, 2020). Unlike traditional games, serious games are designed with specific educational goals, aiming to engage students and enhance their understanding through immersive and engaging experiences (Cooper and Bucchiarone, 2023; de Sousa Borges *et al.*, 2014). Some studies have already proposed serious games to support teaching various software engineering topics, such as project management (Navarro and van der Hoek, 2004) and development processes (Aydan *et al.*, 2017). However, there is still a gap in the use of serious games for teaching software quality models, like *MPS-SW*.

To address this gap, this paper introduces *MPS Manager*, a serious game designed to assist educators in teaching software quality improvement frameworks, focusing on the *MPS-SW* model (SOFTEX, 2021), a Brazilian model developed by Softex[1] and based on *CMMI* model. The goal is for students to implement the model and reach the highest maturity levels. *MPS Manager* incorporates the different maturity levels of *MPS-SW*, allowing students to make decisions and carry out activities corresponding to the appropriate level.

MPS Manager was evaluated with 83 students, divided into four classes from three Brazilian universities. Using the *MEEGA+* model (Petri *et al.*, 2017), a method with a form for evaluating educational games in computing, we analyzed usability and student experience across nine dimensions, including usability, confidence, challenge, satisfaction, social interaction, enjoyment, focused attention, relevance, and perceived learning. Data collection was based on a five-point Likert scale, ranging from “I totally disagree” to “I totally agree” (Likert, 1932).

The game evaluation aimed to verify whether the *MPS Manager* supports the learning process and to identify how different factors affect students’ experiences. Accordingly, this study was guided by the following research questions:

- **RQ1: What are the students’ perceptions regarding the use of the *MPS Manager* to support learning about software quality improvement models?**
- **RQ2: Are students’ perceptions of the *MPS Manager* influenced by characteristics related to their profile or the educational context?**

The empirical results suggest that *MPS Manager* has proven to be a promising platform for teaching *MPS-SW*, with the potential to be adopted in various educational contexts, in-person and remotely.

The main contributions of this work include:

- **Creation of *MPS Manager***: Development of a serious game designed specifically to teach software quality improvement models, focusing on *MPS-SW*.
- **Validation of effectiveness**: An empirical study was conducted to assess the game’s impact on student learning, evaluating its usability, engagement, and educational outcomes.
- **Identification of improvement opportunities**: Analysis of students’ experiences led to practical recommendations to enhance the design and adoption of serious games for teaching software quality and process improvement models.

This article is organized as follows. Section 2 presents important concepts of process quality models and related works. Section 3 introduces the *MPS Manager* game design and describes a usage scenario of the *MPS Manager*, focusing on students’ gameplay. The evaluation method is presented in Section 4. The principal results and findings discussion are presented in Sections 5 and 6 respectively. Finally, Section 7 concludes the work by citing the contributions and providing insights for future directions.

[1] Association for the Promotion of Brazilian Software Excellence – `https://softex.br`

2. Background and Related Works

2.1. *Software Process Quality Models*

Process quality models can be used as guides for companies aiming to enhance the quality of their products through process improvement. These models outline all activities that should be followed in a formal and organized manner, enabling a company to be evaluated and assigned a maturity level that reflects its progress (Sommerville, 2016). The most well-known models include *CMMI-Dev* (Capability Maturity Model Integration for Development) (CMMI, 2021) and the *MR-MPS-SW* (Reference Process Model for Software) (SOFTEX, 2021).

CMMI-Dev is an international model structured into process areas, providing an evolutionary path for companies to improve their software development processes. The 2.0 version consists of six maturity levels: (0) Incomplete, (1) Initial, (2) Managed, (3) Defined, (4) Quantitatively Managed, and (5) Optimizing (CMMI, 2021).

In the Brazilian context, the *MR-MPS-SW* has its specialization, the *MPS-SW*, which was designed to accommodate the needs of micro, small, and medium-sized national software companies, to reduce costs associated with international evaluation and certification processes.

MPS-SW comprises the general model guide and software acquisition and implementation guidelines. *MPS-SW* is used to assess software companies into seven maturity levels (SOFTEX, 2021), as depicted in Fig. 1: (G) Partially Managed, (F) Managed, (E) Partially Defined, (D) Defined, (C) Totally Defined, (B) Quantitatively Managed, and (A) Optimizing, with level G representing the most basic and level A the most advanced. This model evaluates and classifies companies at one of these non-cumulative improvement levels. To advance to higher levels, a company must satisfy all the requirements of the preceding levels, as well as those of the target level.

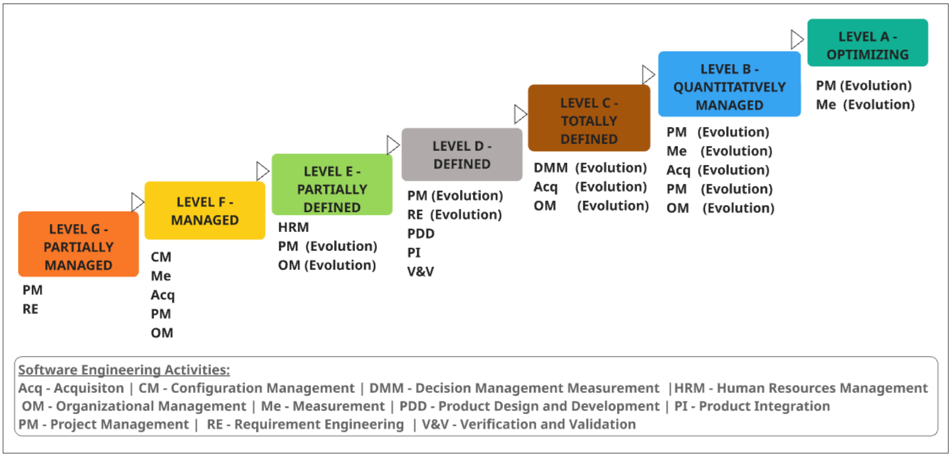


Fig. 1. Process improvement levels in *MPS-SW*. Adapted from (SOFTEX, 2021).

From the 2021 version, processes are organized into project and organizational processes. Each level comprises a combination of processes described in terms of purpose, outcomes, and process capability. Process capability reflects the execution efficiency and progresses with the maturity levels. Therefore, the achievement of each process capability level is assessed based on the corresponding implementation results (SOFTEX, 2021).

2.2. *Gamification and Serious Games*

In recent years, gamification and serious games have been widely adopted in educational contexts as strategies to increase student engagement and enhance learning effectiveness.

Although both concepts share elements from the gaming world, they differ in important ways. Gamification is the use of game techniques and elements in non-game contexts, such as classroom exercises or assessments. Serious games, in contrast, are complete, interactive, and immersive games developed with a purpose beyond entertainment, such as teaching specific content or developing professional skills (Aydan *et al.*, 2017; Bai *et al.*, 2020).

Gamification has been used as an alternative in education to increase student engagement by bringing elements from the gaming world into the educational domain (Martins *et al.*, 2018; de Sousa Borges *et al.*, 2014). Game mechanics, dynamics, and strategies are game elements that enhance learning when used to address a specific problem. Studies suggest that gamification may contribute to improved learning outcomes by promoting enthusiasm, providing immediate feedback, enabling social recognition, and motivating goal pursuit (Bai *et al.*, 2020; Furdu *et al.*, 2017). For instance, avatars and virtual worlds can promote development and organization through self-motivation; rankings and levels can stimulate competition for social recognition among players, and group tasks can foster collaboration among players (Furdu *et al.*, 2017). However, its use requires caution: some students may perceive gamified activities as ineffective or may experience negative emotions such as anxiety or jealousy, depending on individual characteristics (Bai *et al.*, 2020).

While gamification focuses on integrating game elements into existing educational activities, serious games offer a more immersive experience by simulating real-world situations to prepare players for specific scenarios. Serious games are commonly used to support education in various subjects, providing support for game-based learning approaches (Aydan *et al.*, 2017).

For both gamification and serious games to reach their educational potential, it is essential to consider motivational factors that sustain engagement throughout the experience. Elements such as challenge, curiosity, control, cooperation, competition, recognition, feedback, immersion, real-world relevance, and social interaction play a central role in this process (Laine and Lindberg, 2020). When effectively integrated, these factors help students perceive the activity as rewarding, increasing their engagement with the game and their willingness to learn through it (Laine and Lindberg, 2020).

2.3. *Gamification and Serious Games in Software Engineering Education*

The use of gamification and serious games in science education (e.g., mathematics and physics) is not recent, with documented applications dating back to the late 1970s (Ekin *et al.*, 2023). In the field of computer science education, research on serious games has accelerated over the last decade. However, when it comes to software engineering education specifically, the use of gamification and serious games remains in its early stages.

In the secondary study of Alhammad and Moreno (2018) and Barreto and França (2021), the authors concluded that, although gamification shows great promise, many gaps and challenges remain, including the need for more empirical studies to advance research in the field further.

More recent studies characterized serious games in software engineering education. For instance, Furtado *et al.* (2021) and Tonhão *et al.* (2023) reported the use of diversity of game elements (e.g., achievements, badges, leaderboards), dynamics (e.g., narrative, progression), genres (e.g., board, digital, cards, strategy), and mechanisms (e.g., challenges, cooperation, rewards) for improving students engagement in this discipline.

Several interesting initiatives can be found in the context of software development process education. Serious games in this area aim to simulate the software development environment, where students take on roles as project managers and make decisions such as hiring and firing employees (Navarro and van der Hoek, 2004; Kohwalter *et al.*, 2011; Aydan *et al.*, 2017; Moura and Santos, 2018). *SimSE* (Navarro and van der Hoek, 2004) and *Floors* Aydan *et al.* (2017), focus on simulating software development processes through serious games. In those games, students act as project managers to develop software according to a specific process. A more specific game, the *SDM* (Kohwalter *et al.*, 2011), simulates a development team, taking into account human characteristics as well. Some important findings in some of these proposals is that, participants using the serious game tended to have a slightly more positive experience, while those using traditional methods tended to have a neutral experience Aydan *et al.* (2017).

Similarly, Moura and Santos (2018) propose a board game called *ProcSoft* to teach software processes in a more informal and relaxed manner. Through the game, students explore topics related to the *ISO/IEC 29110* standard, learning about the phases that comprise the software development lifecycle. The game's goal is for participants to create a complete software process, considering real-life constraints during software development, such as team size and financial resources. Another non-digital game, the *Ball Point Game*, focuses on continuously improving software development processes. This game aims to enhance project estimations and the efficiency of software teams (Calderón *et al.*, 2019). In the *Ball Point Game*, during each iteration, teams estimate the number of balls (effort) they believe they can pass within a given time frame, both with and without faults. This helps teams empirically assess their performance, improve estimations in the next iterations, and reduce the number of faults they make.

Chaves *et al.* (2015) investigated how the serious game *DesignMPS* improves software process modelling learning. As a result, the authors found that students learning with serious games perform similarly to those using active learning methods, such as project-based learning. Finally, Silva *et al.* (2023) proposed a mobile game to assess students knowledge about *MPS.BR*, however such game is based on question-answer activities. Neither gamification nor simulation on realistic applications of *MPS.BR* was considered in such a game.

To the best of our knowledge, ***no digital game has been identified in the literature with a primary focus on teaching software process quality models***. In this context, the novelty of the *MPS Manager* lies in its ability to teach the *MPS-SW* model, considering its levels in the decision-making process determining the student's game progress. Additionally, the *MPS Manager* incorporates gamification elements, such as medals, leaderboards, and avatars to engage students in learning this important topic through gameplay.

3. MPS Manager

The *MPS Manager* game is intended for university students in technology-related fields and aims to support the teaching and learning of the MPS-SW model. This section presents the *MPS Manager* from five complementary perspectives: (i) game design principles; (ii) user experience (UX) and user interface (UI) elements; (iii) functional design and user roles; (iv) system architecture; and (v) a usage scenario illustrating the application of the game in an educational context. Each of these aspects is detailed in the following subsections.

3.1. *Game Design Principles in the MPS Manager*

The *MPS Manager* game was designed based on the six facets of serious game design proposed by Marne *et al.* (2012): pedagogical objectives, domain simulation, interaction with the simulation, problems and progression, decorum, and conditions of use.

Regarding **pedagogical objectives**, the game was developed to support the first three levels of Bloom's Taxonomy (Starr *et al.*, 2008) – remember, understand, and apply – specifically focusing on helping students learn and apply the MPS-SW model during software project execution. The game aims to foster foundational knowledge of software process improvement in a contextualized and practical manner.

In terms of **domain simulation**, the game is set in a fictional software company seeking to improve its development processes. It is designed to simulate real-world MPS-SW model implementation and optimization scenarios, providing students with a hands-on and interactive learning experience.

Students **interact with the simulation** by taking on the role of process improvement managers. They are responsible for making strategic decisions, such as allocating resources, hiring personnel, and acquiring tools – all aligned with the maturity levels defined by the *MPS-SW* model, as described in Section 2.1. These decisions directly influence the organization's ability to progress through the maturity levels.

The **problems and progression** facet is represented through the game's structure, which comprises seven maturity levels, from G (initial level) to A (final level). At each level, students must manage budget and time constraints to carry out software engineering activities (e.g., PM – project management, RE – requirement engineering, etc.) that directly impact the quality of the implemented processes in the fictional company. Success depends on their ability to correctly perform these activities and to iteratively refine them, ensuring continuous improvement throughout the simulation. To increase realism, the game introduces unexpected challenges, such as license shortages or staff limitations, which require students to adapt their strategies and iteratively improve their processes.

The ultimate goal is to reach level A of the *MPS-SW* model in no more than ten semesters, simulating a complete process improvement cycle. Although this timeframe is arbitrary, it was chosen to divide the game into manageable periods for gameplay purposes. During the game, students will face incidents and challenges inspired by real-world situations – such as team management failures or budget constraints – requiring them to make strategic decisions to overcome obstacles and progress through the maturity levels. Managers (students) can request periodic assessments, simulating the real audit process of the *MPS-SW*, where a fictional evaluation institution verifies the progress of process implementation.

Regarding the **decorum facet**, *MPS Manager* adopts a clean and accessible visual style, with a light color palette, intuitive icons, and clearly structured panels. These elements represent key aspects of the simulation, such as resource availability, team composition, progress indicators, and unexpected events (see Fig. 2 and Fig. 8). The interface

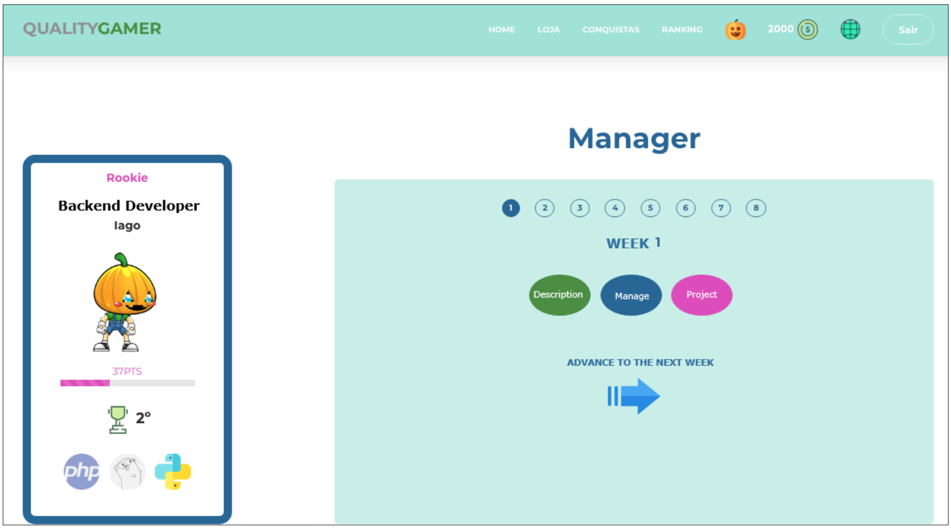


Fig. 2. Home interface of the *MPS Manager*.

emphasizes clarity and ease of use, helping students focus on strategic decision-making. Features such as project delay indicators, incident alerts, and a visual maturity staircase (see Fig. 1) provide feedback and contribute to player immersion and engagement with the process improvement flow.

Finally, in terms of **conditions of use**, *MPS Manager* was designed to be used in two instructional settings: (i) synchronously in the classroom, with instructor support, or (ii) asynchronously, allowing students to progress independently while receiving guidance through complementary materials or remote feedback. This flexibility supports diverse learning contexts and facilitates the integration of the game into different course formats.

3.2. *UX/UI Elements in the MPS Manager*

The user experience (UX) and user interface (UI) of the *MPS Manager* game were carefully designed to enhance usability, engagement, and learning effectiveness. The following design principles were applied:

Clarity and Feedback

The game provides clear visual feedback to guide the user, as depicted in Fig. 2. Action buttons like *"Advance to the next week"* and context-specific actions like *"Develop"*, *"Manage"*, and *"Project"* use distinct colors and are spatially separated, making their functions immediately recognizable. A progress tracker at the top of the screen with numbered circles (e.g., *WEEK 1*) reinforces the sense of structure and advancement.

Visual Hierarchy

Key interface elements – such as the current weekly action and the *advance* button – are visually emphasized and centrally positioned. The screen layout is divided into intuitive zones: the left column displays the character avatar and overall progress, while the right column presents weekly actions. This layout supports natural navigation and reinforces user orientation.

Consistency

Icons, colors, and typography follow a unified visual language throughout the game. The interface maintains a cohesive style, with rounded buttons, a consistent color scheme, and playful yet readable typography. This consistency fosters familiarity and ease of use.

Gamification Elements

To increase engagement, the game integrates classic gamification components, such as a character/avatar, progress bar, skill levels (e.g., Python, PHP), and titles (e.g., Rookie). The week-by-week progression mirrors a typical level-up mechanic found in games, maintaining player motivation and a sense of achievement.

User Control and Freedom

The user has clear control over gameplay pacing through the *"Advance to the next week"* button. All available actions are visible and accessible at any time, avoiding hidden options and supporting user autonomy and decision-making.

Minimalism

Each screen presents only essential information – avatar, status, current week, and available actions – avoiding distractions and reducing cognitive load. This minimalist approach supports focus and contributes to a smoother learning experience, particularly in educational settings.

Visual Metaphors

The interface employs a friendly, illustrative visual style. Cartoon-like icons, badges, keys, locks, medals, and energy bars serve as intuitive visual metaphors that convey progress, access, and status. These playful visual cues help users quickly understand game mechanics and reinforce the concept of learning through play.

3.3. *Functional Design and User Roles of the MPS Manager*

This section presents the functional design of the *MPS Manager*, focusing on the main user roles and their interactions with the system's core features. The objective is to clarify how the behavior of the game aligns with its educational goals, allowing instructors to configure simulations and students to engage with realistic scenarios of process improvement.

The use case diagram presented in Fig. 3 provides an overview of the game's main features, detailing the interactions between the two primary stakeholders: the student, who assumes the role of process improvement manager, and the professor, who oversees the model, activities, and challenges associated with each level.

In the following subsections, we detail the game's features, explaining how the user profiles – professor and student – interact with the process model and demonstrate the application of these features in a practical usage scenario.

3.3.1. *Professor's User Profile*

The professor is responsible for managing classrooms and has total flexibility to add and customize models in *MPS Manager*. By default, the *MPS-SW* and its levels come pre-configured in the game, providing an initial structure for process improvement simulations based on this model. However, the professor can also configure other models, adjusting them to meet the specific objectives of their course, as explained in Section 3.3.4.

After authentication, the professor can set up classrooms, choosing between a custom model or the default model available in the system. The "Manage Classroom" use case allows the professor to create challenging classrooms, adapting the content and models according to the class's needs.

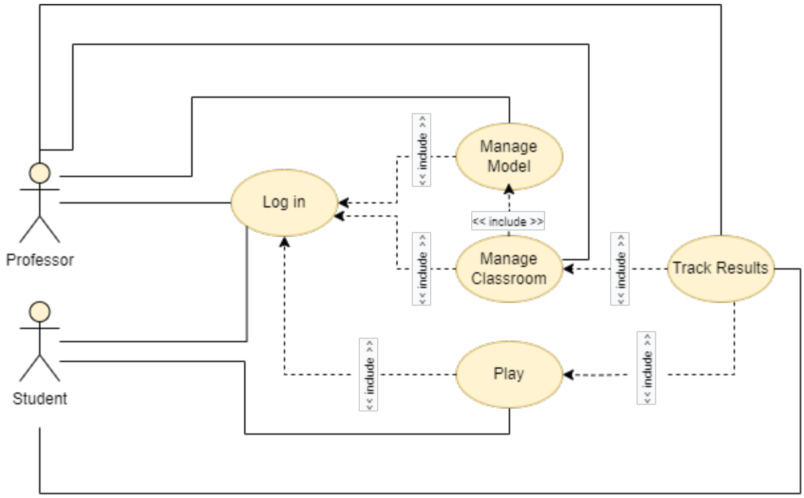


Fig. 3. Use cases diagram for the *MPS Manager* serious game.

Fig. 4. Description of the project context in the *MPS Manager* serious game.

Additionally, the professor can associate a project description through which students will simulate the implementation of the model used in the class, allowing them to view this information. Fig. 4 presents an example of a project description in *MPS Manager*. The “Track Results” use case enables the professor to monitor student performance throughout the game, tracking their results, and the teacher can communicate with students through the platform.

3.3.2. *Student’s User Profile*

The student can play *MPS Manager* by joining the class created by their professor, as represented by the use case “Play”, shown in Fig. 3. Upon accessing the *MPS Manager*, the student can view the project assigned by the professor and either start the game or continue from where they left off if they have already begun.

As previously explained, the student must make informed decisions regarding the processes related to the current level. If the game registers any incident or issue, the student must think strategically and make decisions to adjust resource management and solve these problems. The student can choose to advance to the next level at any point.

When this happens, the game will calculate which processes should be achieved or improved without directly notifying the student. Then, the game will display an evaluation of the management so far and the current company level controlled by the student. If the student does not reach level A by the end of the ten semesters, they will receive a score based on the level they achieved.

The student also has access to the class ranking, where they can view their performance compared to their peers and interact with other students through a chat system.

3.3.3. *Manage Model Use Case*

The manage model use case consists of a set of rules and levels that guide the game's decision-making algorithm. In this use case, the game is responsible for managing the MPS-SW model and its structure, including levels, processes, activities, resources, incidents, and their respective resolutions.

The quality model is fully configurable by the professor. As illustrated in Fig. 5, the *Model* consists of maturity levels corresponding to the different stages of process evolution. The game's default pre-configured model is *MPS-SW*, which consists of seven levels. However, the professor has the flexibility to adjust not only the levels but also other components of this model, such as activities, resources, and processes, according to the teaching goals of the course. Each *Level* is associated with specific processes that need to be implemented or optimized by the player. Each *Process* contains *Activities* that require *Resources* to be executed, and both have assigned *Scores*. Thus, the professor can define, for each level, which processes are part of it and which activities and resources will be needed for their execution.

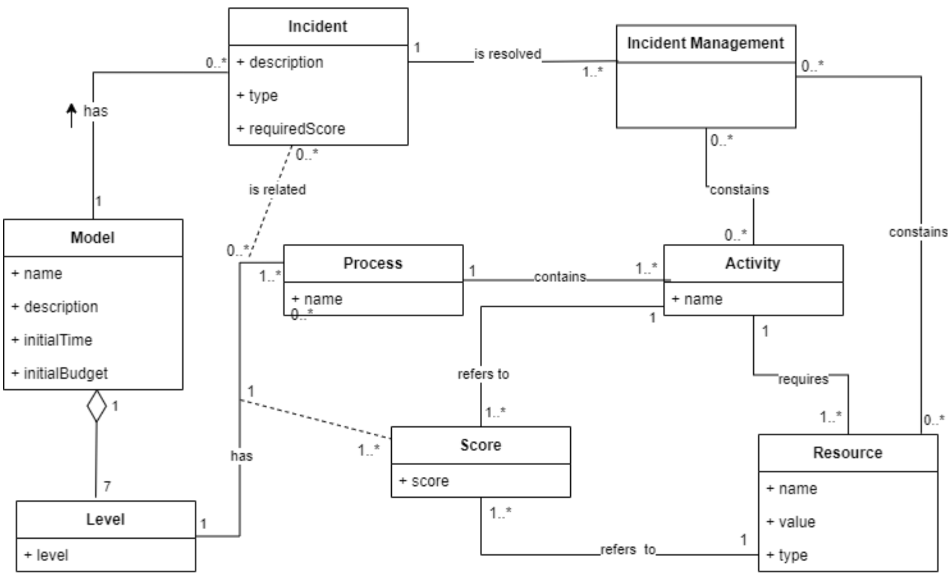


Fig. 5. Conceptual model of the *MPS Manager* serious game.

The game uses the scores assigned to the processes and resources to internally calculate the student's progress, although these scores are not visible to the student. This component sets a scoring limit for each resource or activity, preventing students from focusing all their budget or time on a single management aspect.

During the game, *Incidents* may occur randomly or as a consequence of incorrect decisions made by the student when managing the processes at a given level. To resolve these incidents, the student must perform one or more specific actions in *Incident Management*, achieving the required score to fix the problem and continue advancing in the model. This aspect of the game challenges the student to reflect on the focus they should be given to processes at each level and how they are divided and matured as the company progresses through maturity levels.

3.3.4. *Play Use Case*

The "Play" use case is responsible for the game's dynamics and controls the student's progress through a decision tree, as shown in Fig. 6. In each play, the system evaluates the student's decisions based on the scores obtained, which can result in different scenarios: advancing to the next level (represented by the green circle in the decision tree) if all decisions are correct and all incidents are resolved; remaining at the current level with new incidents (represented by the red circle in the decision tree), if wrong decisions are made, resulting in the generation of new incidents; or remaining at the current level without new incidents (represented by the gray circle in the decision tree), if correct decisions are not enough to advance due to unresolved previous incidents. Additionally, the system may randomly generate incidents over which the student has no control but must resolve to ensure progress in the model.

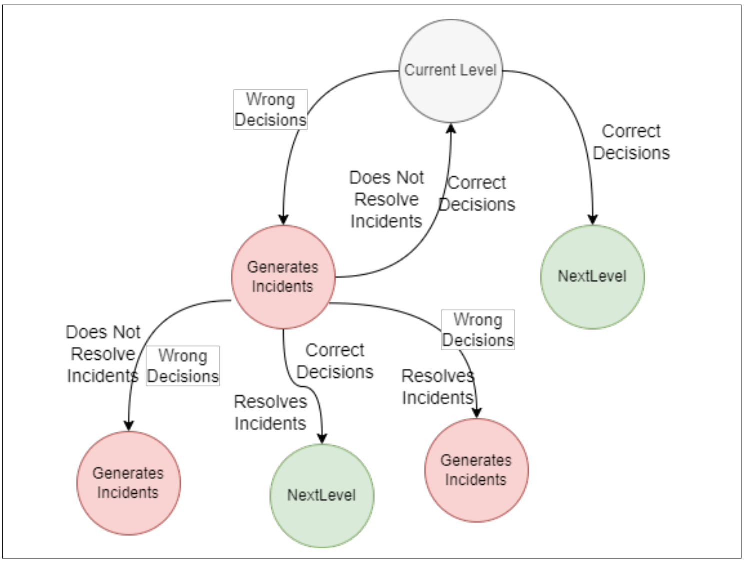


Fig. 6. Decision tree of scenarios performed by the player ,

3.4. *System Architecture of the MPS Manager*

The *MPS-SW* game was developed using a microservices architecture to ensure the system is scalable and adaptable for future needs. As illustrated in Fig. 7, the system is composed of two main microservices – *User* and *Manager* – orchestrated via an *API Gateway* that mediates all communication between the frontend and backend components.

This architectural choice offers several advantages:

- **Ease of Deployment**: Each microservice can be independently deployed and updated, reducing the risk of regressions and allowing targeted maintenance and faster iteration cycles.
- **Fault Isolation via Circuit Breaker Pattern**: Failures in one service (e.g., the Manager service) do not compromise the availability of others. This isolation mechanism enhances system robustness and fault tolerance.
- **Modularity and Reusability**: The services were designed following principles of low coupling and high cohesion, supporting their reuse across different courses, contexts, or extensions of the game.

The *User* microservice handles authentication, user profiles, session tracking, and access control, ensuring that students can only access their respective classes and game progress. Meanwhile, the *Manager* microservice is responsible for core game logic, including model configuration, level progression, resource and activity handling, and scoring.

Security was considered as part of the architectural foundation. The authentication mechanism ensures that only registered users can access the system, while access control mechanisms restrict users from interacting with unauthorized resources or classes.

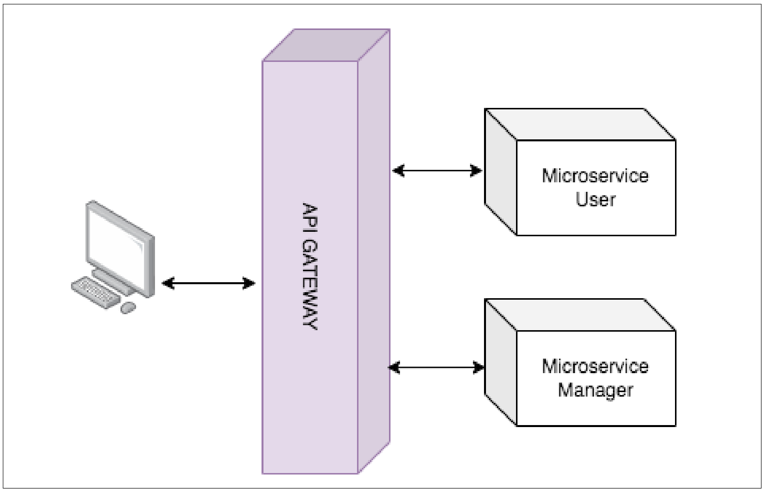


Fig. 7. Microservices-based architecture of the *MPS-SW*, composed of an API Gateway and two core services: *User* and *Manager*.

This modular and resilient architecture enables the system to support multiple simultaneous classrooms and students, offering an infrastructure suitable for both small-scale academic use and broader institutional deployment.

3.5. *Scenario of Use of MPS Manager*

In this section, an example of using the *MPS Manager* game will be considered, simulating possible interactions between the professor and the student with the game.

The first step for the professor is to choose the model that will be used in the class. As mentioned earlier, the *MPS-SW* is pre-configured by default, but the professor has the flexibility to modify it or even create their own model. If the professor opts to configure the model, they can create maturity levels and assign scores to activities and resources based on their relevance to each level. Table 1 provides an example of configurations for Level G of the *MPS-SW*, where higher scores are assigned to activities directly aligned with the objectives of that maturity level, such as “Project Management” and “Requirements Engineering”, encouraging students to focus on the most relevant processes.

In the example presented in Table 1, for the “Requirements Engineering” process, the professor configured the activity “Perform requirements gathering” along with the resource “Requirements Analyst”, assigning them scores of 10 and 15, respectively. Additionally, the professor included other resources such as “Programmer” and “IDE License”, which received lower scores of 2 each. While these resources are essential for the company’s development, they are not critical to the specific goals of this level. Resources like “Designer”, “Tester”, and “UI Software License” are not necessary for Level G and, therefore, are assigned negative scores (-2), reflecting their lack of relevance and posing challenges for students if they are selected incorrectly.

These model configurations can be applied directly in the classroom and saved for future reuse if the professor wishes to use them in other classes. Next, the professor creates the class and provides a description of the project to be worked on. The description

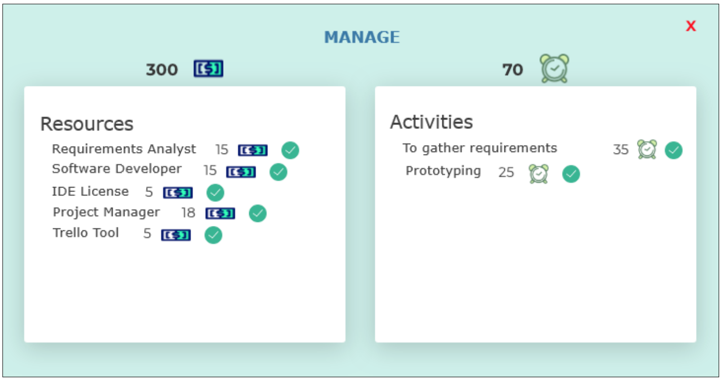
Table 1
Example of Model Settings in MPS Manager

Level	Process	Activity / Resource	Score
G	Project Management	Detailed project planning	10
G	Project Management	Project Manager	15
G	Project Management	Trello®	10
G	Requirements Engineering	Perform requirements gathering	10
G	Requirements Engineering	Requirements Analyst	15
G	Other	Programmer	2
G	Other	IDE license	2
G	Other	Designer	-2
G	Other	Tester	-2
G	Other	UI Software License	-2

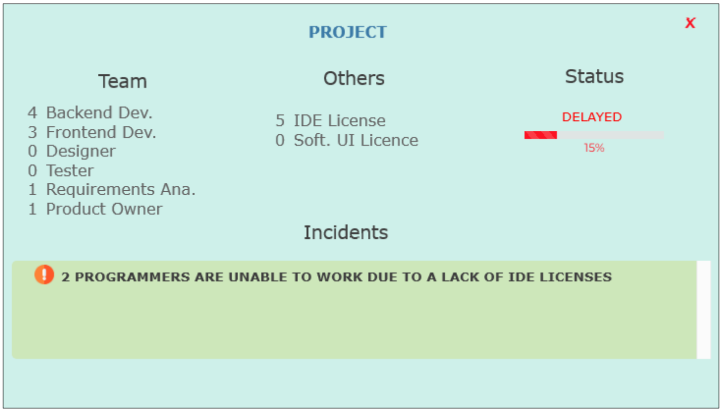
in the example shown in Fig. 4 refers to a software development project for a hotel reservation. This description helps students quickly engage with the content and understand the context they will be immersed in, facilitating the application of their learning in a simulated environment.

This setup prepares students for making strategic decisions in the “Manage” tab, where they allocate resources and plan activities to progress through the levels, aiming to reach Level A within ten semesters. Students start in the first semester of the project implementation phase, receiving a budget to acquire resources and a set amount of time to carry out activities, as previously configured by the professor.

In the first semester, it’s recommended that the student, acting as the company manager, allocates their resources and time to positions and activities that contribute to “Project Management” and “Requirements Engineering”, resulting in higher scores. Positions such as “Project Manager”, “Requirements Analyst”, and tools like “Trello®” are recommended for this initial period. The game begins in the “Home” tab, as shown in Fig. 2. From there, the student can access the “Manage” tab (Fig. 8 (a)), where they can



(a) Management interface



(b) Project interface

Fig. 8. *MPS Manager* Interfaces.

hire staff and allocate time according to the company's maturity level, aiming to reach Level A within ten semesters.

By understanding the concepts of this level, students are expected to invest in roles such as "Requirements Analyst", "Project Manager", and "Programmer" to reach Level G. It is also essential for them to dedicate their time to the activity "Perform requirements gathering". As time progresses, the game will evaluate the student's decisions, determining if they have reached Level G, provided their choices align with the standards of that level. This progress can be tracked in the "Project" tab, as illustrated in Fig. 8(b).

On the other hand, consider a student who does not fully grasp the concepts and decides to hire too many "Programmers", consuming a significant portion of the budget on staff and leaving insufficient resources for essential activities and tools, such as hiring a "Project Manager" or purchasing an 'IDE licence' for the team. As a result, it will not be possible to advance to the next level over time, and this decision will trigger an incident. Fig. 8(b) presents an example of an incident caused by the failure to purchase an IDE for the programmers.

In the project interface, the student can track the progress of the model implementation project, view the company's current team, and identify any incidents that need to be solved.

4. MPS Manager Evaluation

The **main goal** of this evaluation is to carry out an empirical study to identify participants' satisfaction regarding usability and user experience by playing the *MPS Manager* serious game.

To achieve this objective, the *MPS Manager* game was planned to be used as an activity in software engineering disciplines of undergraduate courses. Specifically, this game was applied in four disciplines taught at three different Brazilian public universities located in different states, i.e., Fluminense Federal University (UFF), Federal University of Grande Dourados (UFGD), and Itajubá's Federal University (UNIFEI). To execute the evaluation, students were recruited from the same institutions as the professors responsible for the disciplines, using convenience sampling (Alkassim *et al.*, 2016).

Table 2 summarizes the real scenarios where the *MPS Manager*'s assessment had place.

Table 2
Courses characteristics for *MPS Manager* evaluation

University	Course	Discipline	Previous Training	Advanced	In-person
UFF	Information Systems	Software Quality and Testing	No	Yes	34
UNIFEI	Information Systems	Software Engineering II	Yes	Yes	24
UFGD	Information Systems	Software Engineering I	Yes	No	0
	Computing Engineering	Software Engineering I	Yes	No	11

4.1. *UFF's Scenario*

At UFF, the evaluation took place in "Software Quality and Testing", an advanced discipline taught to Information Systems students, a nighttime undergraduate course[2]. Thirty-four students participated in-person, while seven students completed the activities online.

The game was used in the "Standards and Process Quality Models" topic, focusing on Software Quality Improvement. The professor conducted an informative lecture on the *CMMI* framework in the corresponding class, lasting four hours. Students were oriented to assist a video class of the *MPS.BR* framework. After that, the *MPS Manager* game was presented to students, explaining its functioning. As an extra class activity, the professor encouraged the students to play and assess the game.

4.2. *UNIFEI's Scenario*

At UNIFEI, the *MPS Manager* was assessed in the context of the "Software Engineering II", and advanced discipline as part of the Information Systems course, offered to 24 undergraduate students in the modality in-person and nighttime.

The game was used in the "Software Process Improvement (PSI)" topic. Before this topic, students had already been exposed to topics related to Software Development Processes. Students were introduced to the generic *PSI* framework and therefore, *CMMI*, *SPICE*, and *MPS.BR* were explained by the professor. In total, all class sessions spanned eight hours. In an additional class, the professor presented the *MPS Manager* game to students. As an in-class activity, the students played the game in the classroom, and after that, they were oriented to answer the form to assess the game.

4.3. *UFGD's Scenario*

At UFGD, the *MPS Manager* game was applied in two undergraduate courses. The first evaluation was conducted by seven students of the Information Systems course in the "Software Engineering I" discipline, offered in the evening as an extra-class activity in an online format. The second evaluation was conducted by 11 students from the Computing Engineering course, offered in-person and in a daytime modality during the "Software Engineering I" discipline. In both instances, the professor responsible for the disciplines trained students in software processes, process quality improvement, and the *MPS.BR* framework topics. Those training sessions were organized to last a total of eight class hours.

---

[2] A nighttime course is a program that takes place during the evening or night hours, typically after regular working hours – often starting around 6 PM or later.

In the Computer Engineering course, the *MPS Manager* game evaluation occurred as an in-class activity, where students played the game in the classroom, and afterward, they were oriented to answer the game's form assessment.

On the other hand, in the Information Systems course, the evaluation was executed as an out-of-class activity, where the professor motivated the students to play the *MPS Manager* game at home, and after that, they were oriented to answer the form to assess the game.

4.4. *Disciplines Syllabus*

The *Software Engineering I* disciplines aim to introduce preliminary concepts of software engineering such as software processes (e.g., iterative, incremental, agile), the software quality concept, requirements engineering, and a brief introduction to *SPI*.

The *Software Quality and Testing* and *Software Engineering II* are advanced disciplines, aimed to provide a comprehensive and deeper understanding of the software development process, emphasizing process and product quality models, *SPI*, as well as introducing concepts about software testing techniques, levels, and automation.

4.5. *MPS Manager Evaluation Execution*

The *MPS Manager*'s evaluation by students was conducted as follows. Once the theoretical background of MPS had been presented to all students, the *MPS Manager* game and the goal of the assessment were introduced to them. They were invited to play the scenario set by the professors, as explained in Section 3.5. Student volunteers read and agreed to the Free and Informed Consent Terms.

Thereafter, the students were invited to answer an online form to assess the usability and user experience. Such a form was based on *MEEGA+*, a method proposed by Petri *et al.* (2017), which is used for educational games evaluation in the computing education context. The model offers pre-defined and tested questions to assess serious games during the construction and testing stages. In *MEEGA+*, the following dimensions are proposed: **1. Usability**; **2. Confidence**; **3. Challenge**; **4. Satisfaction**; **5. Social Interaction**; **6. Enjoyment**; **7. Focused Attention**; **8. Relevance**; and **9. Perceived Learning**. Moreover, the usability dimension is divided into 5 sub-dimensions, namely, **learnability, operability, aesthetics, accessibility, and user error protection**. Each dimension/sub-dimension is specialized in qualitative assessment items. The *MEEGA+* dimensions/sub-dimensions and items are detailed in Table 3. In addition, demographic items were added to the form aiming for a better characterization of student participants of the *MPS Manager* evaluation. Finally, two form items were included to know students' opinions and suggestions about the *MPS Manager*. The assessment form was created using *Google Forms* following the *MEEGA+* form structure available in (Petri *et al.*, 2019).

Table 3

Dimensions used to evaluate the *MPS Manager* game. Adapted from (Petri *et al.*, 2019)

Dimension	Sub-Dimension	Form Item	Results summary
Usability	Aesthetics	The game design is attractive.	mode = 4; median = 3; mean = 3.12; $\sigma = 1.15$
	Aesthetics	The text font and colors are well blended and consistent.	mode = 4; median = 4; mean = 3.75; $\sigma = 1.01$
	Learnability	I needed to learn a few things before I could play the game.	mode = 4; median = 3; mean = 3.16; $\sigma = 1.16$
	Learnability	Learning to play this game was easy for me.	mode = 4; median = 4; mean = 3.41; $\sigma = 1.15$
	Learnability	I think that most people would learn to play this game very quickly.	mode = 2; median = 3; mean = 2.87; $\sigma = 1.23$
	Operability	I think that the game is easy to play.	mode = 3; median = 3; mean = 3.37; $\sigma = 1.14$
	Operability	The game rules are clear and easy to understand.	mode = 4; median = 3; mean = 3.22; $\sigma = 1.2$
	Accessibility	The fonts (size and style) used in the game are easy to read.	mode = 5; median = 4; mean = 4; $\sigma = 1.12$
	Accessibility	The colors used in the game are meaningful.	mode = 4; median = 4; mean = 4.19; $\sigma = 0.83$
Confidence	–	The contents and structure helped me to become confident that Iwould learn with this game.	mode = 3; median = 3; mean = 3; $\sigma = =1$
Challenge	–	This game is appropriately challenging for me.	mode = 4; median = 4; mean = 3.4; $\sigma = 1.13$
	–	The game provides new challenges (offers new obstacles, situations or variations) at an appropriate pace.	mode = 4; median = 3; mean = 3.11; $\sigma = 1.16$
	–	The game does not become monotonous as it progresses (repetitive or boring tasks).	mode = 2; median = 2; mean = 2.41; $\sigma = 1.07$
Satisfaction	–	Completing the game tasks gave me a satisfying feeling of accomplishment.	mode = 4; median = 4; mean = 3.22; $\sigma = 1.16$
	–	It is due to my personal effort that I managed to advance in the game.	mode = 4; median = 4; mean = 3.53; $\sigma = 0.97$
	–	I feel satisfied with the things that I learned from the game.	mode = 4; median = 4; mean = 3.4; $\sigma = 0.9$
	–	I would recommend this game tomy colleagues.	mode = 4; median = 3; mean = 2.94; $\sigma = 1.14$
Social Interaction	–	I was able to interact with other players during the game.	mode = 1; median = 2; mean = 2.45; $\sigma = 1.43$
	–	The game promotes cooperation and/or competition among the players.	mode = 1; median = 2; mean = 2.48; $\sigma = 1.39$
	–	I felt good interacting with other players during the game.	mode = 3; median = 3; mean = 2.71; $\sigma = 1.23$
Enjoyment	–	I had fun with the game.	mode = 4; median = 3; mean = 3.07; $\sigma = 1.17$
	–	Something happened during the game (game elements, competition, etc.) which made me smile.	mode = 3; median = 3; mean = 2.88; $\sigma = 1.24$

Continued on next page

Table 3 – continued from previous page

Dimension	Sub-Dimension	Form Item	Results summary
Focused attention	–	There was something interesting at the beginning of the game that captured my attention.	mode = 4; median = 3; mean = 2.99; σ = 1.19
	–	I was so involved in my gaming task that I lost track of time.	mode = 1; median = 2; mean = 2.08; σ = 1.12
	–	I forgot about my immediate surroundings while playing this game.	mode = 1; median = 2; mean = 2.22; σ = 1.14
Relevance	–	The game contents are relevant to my interests.	mode = 4; median = 4; mean = 3.55; σ = 1.1
	–	It is clear to me how the contents of the game are related to the course.	mode = 5; median = 5; mean = 4.34; σ = 0.85
	–	This game is an adequate teaching method for this course.	mode = 4; median = 4; mean = 3.96; σ = 1.04
	–	I prefer learning with this game to learning through other ways (e.g. other teaching methods).	mode = 4; median = 3; mean = 3.29; σ = 1.25
Perceived Learning	–	The game contributed to my learning in this course.	mode = ; median = ; mean = ; σ =
	–	The game allowed for efficient learning compared with other activities in the course.	mode = 4; median = 4; mean = 3.41; σ = 1.01
	–	The game contributed to reinforce the concepts about *MPS-SW*.	mode = 4; median = 4; mean = 3.94 ; σ = 0.82
	–	The game contributed to understanding the maturity levels of *MPS-SW* and its processes.	mode = 4; median = 4; mean = 3.88; σ = 0.92

Table 4

Additional questions in the assessment form

Question		Form Item
Students' gaming profile	-	containing questions about participants' age range, genre, digital gaming (mobile, PC, video, or console games) frequency, and non-digital gaming (cards or board games) frequency.
Students' comments	Positive Comments	Opinions about what did like at the *MPS Manager* game, as open questions.
	Improvements	Suggestions for improving the *MPS Manager* game, as open questions.

Participants answered the statements in Table 3 using the Likert scale (Likert, 1932):

(1) Totally disagree; (2) Disagree; (3) Neutral; (4) Agree; and (5) Totally agree.

It is intended the analysis of these dimensions contributes to evaluating the effectiveness and efficiency of the learning provided by the *MPS Manager* game. Results of this evaluation are presented in Section 5.

5. Results

A total of eighty-three students participated in the evaluation of the *MPS Manager* game. Section 5.1 provides a detailed overview of the demographic characteristics and participant profiles, offering insights into the composition of the study sample.

To answer the research questions, we analyzed participants' responses to statements associated with each evaluated dimension using a Likert scale. For clarity, the percentage analysis focuses only on responses that indicated agreement or disagreement, excluding "neutral" responses to avoid ambiguity in interpretation. However, visualizations such as charts include all response options to provide a comprehensive view of the distribution.

Regarding Research Question 1 (RQ1), the majority of dimensions received positive evaluations. This suggests that the MPS Manager significantly supports students in understanding and learning software quality improvement models, particularly those aligned with the MPS.BR framework. The evaluated dimensions – usability, confidence, challenge, satisfaction, social interaction, enjoyment, focused attention, relevance, and perceived learning – are discussed in detail in Sections 5.2 through 5.11.

Research Question 2 (RQ2) examines whether students' perceptions were influenced by individual and contextual factors, such as course modality, institution, and prior experience with digital games. The impact of these variables on engagement and understanding is explored in Section 5.12.

5.1. *Characterization of Volunteers' Profile*

Fig. 9 summarizes the profile of the research participants in terms of age and gender. In this study, most respondents were aged between 19 and 28 (92%), with a predominance of male students (89%). The total percentage does not equal 100% because the form included the response option "Prefer not to answer". These results reflect the reality of the trend in Computer Science courses in Brazil, indicating that women are a minor-

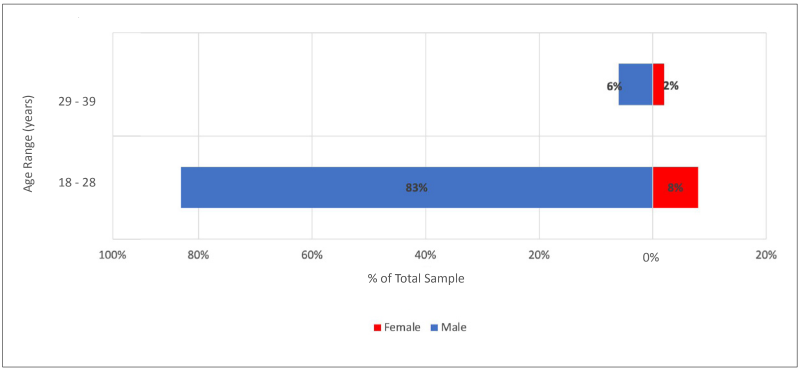


Fig. 9. Participants – Gender and Age Disaggregation.

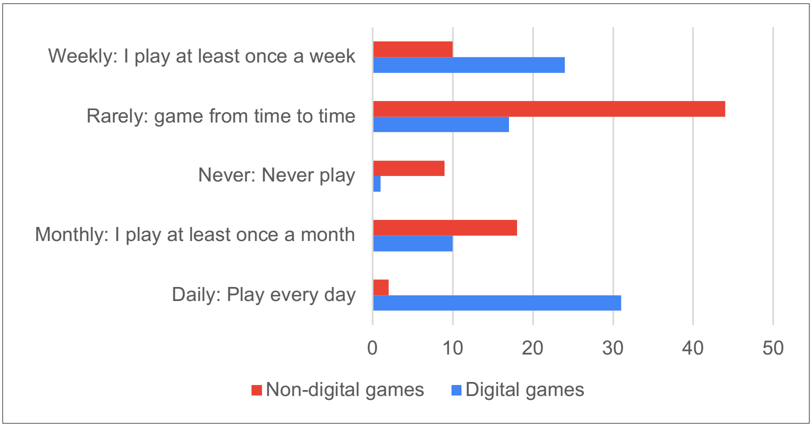


Fig. 10. Participant game play frequency.

ity among graduates of Computer Science courses in the country (Ribeiro *et al.*, 2019; Holanda *et al.*, 2022).

Another characteristic considered in the analysis of the participant's profile was the time spent playing digital and non-digital games. The purpose of analyzing this factor is to assess participants' gaming experience level.

Evaluating the time spent playing digital games, as shown in Fig. 10, most of the participants (37%) spent time every day playing some digital games. Nearly 29% played

at least once a week. Other 21% rarely play games (just from time to time), and 12% spent time playing at least once a month. Only 1%, never play this kind of game. On the other hand, in non-digital games, the majority of participants (53%) rarely game (just from time to time). Others 22% play only one time a week, 11% never play, and 2% play non-digital games daily.

5.2. *Usability Perception*

Fig. 11 shows the results of the usability dimension. Regarding the game's aesthetics, most students (58%) agree that the game colors, design, text, and fonts are appealing. In terms of learnability (ease of learning how to play the game), students also rated the game positively, with the evaluation hovering around (46%) agreement and (36%) disagreement for this sub-dimension. The operability factor received approximately (46%) agreement against (29%) disagreement, suggesting that the game generally has features that facilitate its operation and control. In terms of accessibility, the average agreement score was around (81%), indicating that people with mild to moderate visual impairments could use the game.

Trying to obtain a better understanding of the results, we consider the analysis of the open-question (*RQ11* and *RQ12*) responses. The majority of participants (51%) who answered the open question agreed that they liked the game's aesthetics related to

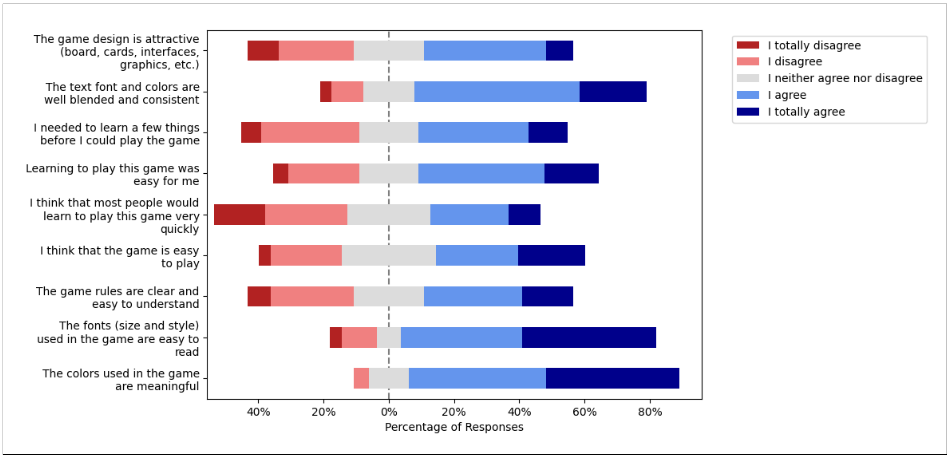


Fig. 11. Students' usability perception.

simplicity (participants *PT-7, PT-13, PT-17, PT-18, PT-21, PT-28*); design (participants *PT-6, PT30*); theme (Participant *PT-4*); avatars or characters (participants *PT-13, PT-24, PT-32*); colors (participants *PT-32, PT-48, PT-61*); and the game interface (participants *PT-39, PT-46, PT-52*).

5.3. *Confidence Perception*

The next dimension evaluated is confidence, as shown in Fig. 12. Based on the analysis of the responses, an average of 36% of the participants agree that the game's content and structure can help them progress in their studies, while 31% disagree. Analysis of the open-ended question regarding perceived improvements for the game (*RQ11*) suggests that issues with the game's structure and feedback may have influenced the inconclusive results mentioned by participant *PT-28*: *"There could be scenarios or stories between the years and feedback on the player's choices. In addition to improving the visibility of level progression"*. *PT-35* stated: *"It's hard to know if I'm doing badly or well, the score could be per round"*.

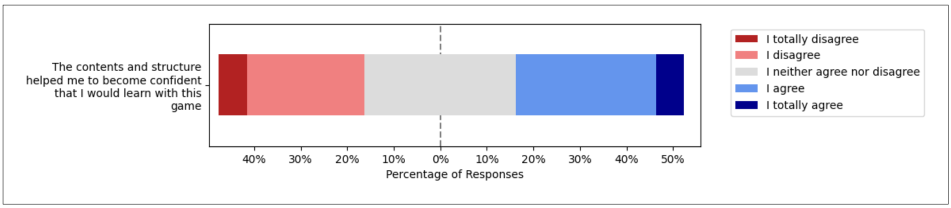


Fig. 12. Students' confidence perception.

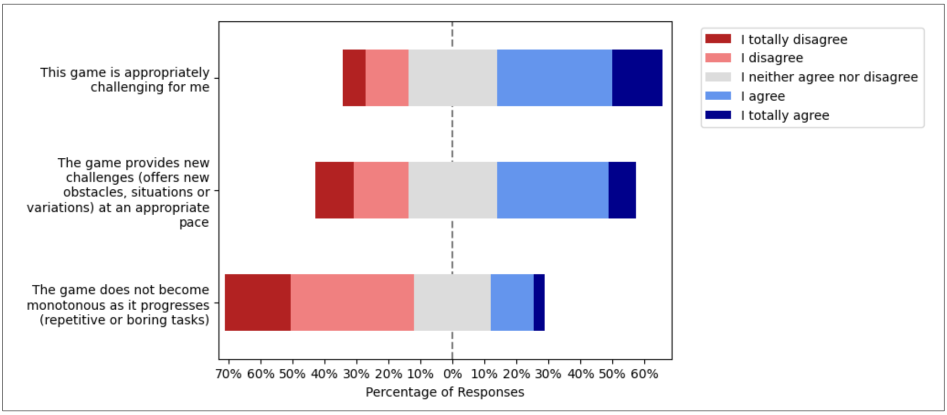


Fig. 13. Students' challenge perception.

5.4. *Challenge Perception*

The evaluation of the study took into account the level of difficulty of the game, which is shown in Fig. 13. On average, about 43% of the students agreed that the game was appropriately challenging, while only 29% disagreed. This suggests that the inclusion of new obstacles and situations as the game progresses helps to maintain interest. Students' responses to the open-ended questions supported these findings by revealing their favorite aspects of the game. *PT-22* stated that he/she liked: *"trying to reach level A"*. Similarly, *PT-23* emphasized the game's: *"competitiveness and trying to reach the maximum level"*, as noted by *PT-44*: *"the competitive part and the ranking"*.

However, focusing solely on the final question of the game can become monotonous. According to most students (59%), the tasks become boring and repetitive over time, which can contribute to dissatisfaction and distraction during the game. These conflicting findings are supported by the participants' comments on requested game improvements. According to *PT-15*: *"It's repetitive"*. *PT-49* said: *"It's necessary to use a more attractive UI"*, and *"Get a loading screen because it takes time to load and we can't see it (PT-61)"*.

5.5. *Satisfaction Perception*

Satisfaction is also a positive factor for the game, as shown in Fig. 14. The results show that, on average, 52% of students feel that the effort they put into the game leads to learning. These results were corroborated by participants' statements when asked to identify positive aspects of the game. The participants pointed out several examples. *PT10* and *PT-15* stated that *"the sense of accomplishment as passing the level"*. *PT-31* also wrote: *"What I liked the most was the ability to try several times and learn from your mistakes."*

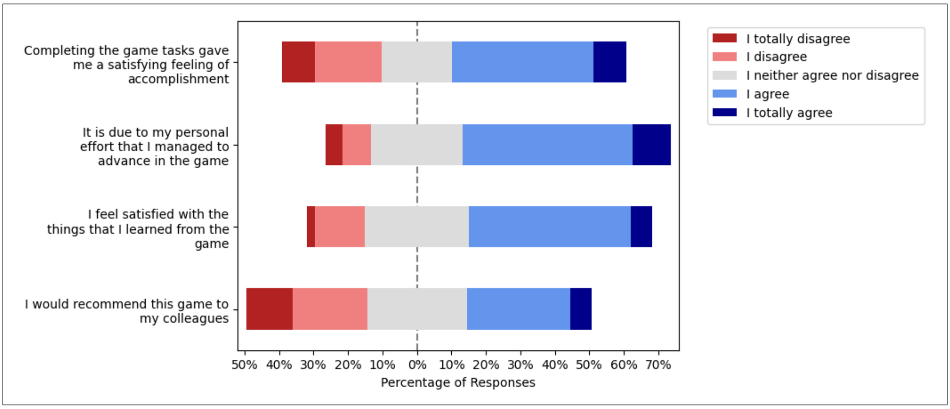


Fig. 14. Students' satisfaction perception.

5.6. *Perceived Social Interaction*

The social interaction factor received a negative rating, as shown in Fig. 15. About 53% of the students disagreed that the game promotes a shared environment for cooperative or competitive activities. This result reflects the fact that the game is a single-player game without interactive chat. In response to open-ended questions about possible game improvements (*RQ11*), participants suggested: *"adding mechanisms for participants to interact (PT-60)"* and *"creating an interactive chat (PT-50)"*.

5.7. *Enjoyment Perception*

Students responded positively to the enjoyment factor, with an average of 40% agreeing that they had fun playing the game, as shown in Fig. 16. When asked what they liked about the game, participants stated: *"Fun way to learn the content (PT-1)"* and *"The avatar moves and it's funny (PT-47)"*.

5.8. *Perceived Focussed Attention*

Fig. 17 presents the results for the focused attention factor. The analysis of the results shows that the majority of students (64%) did not feel able to maintain their concentration and attention during the game, mainly because they did not feel that the game presented interesting elements that captured their attention. However, they did not feel engaged during the game. This finding may have been influenced by the response time and dynamics of the game, as cited by participant *PT-32*: *"Make it more dynamic with activities that do not just involve choosing between pre-determined options."*. In addition, participants identified improvements to the game, such as *"Behavior towards time*

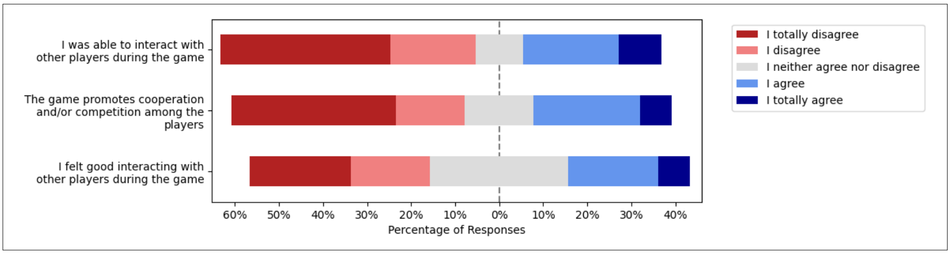


Fig. 15. Students' social interaction perception.

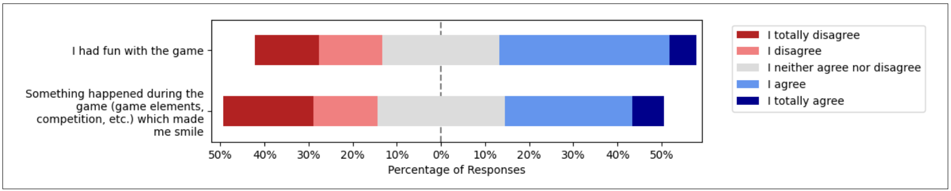


Fig. 16. Students' enjoyment perception.

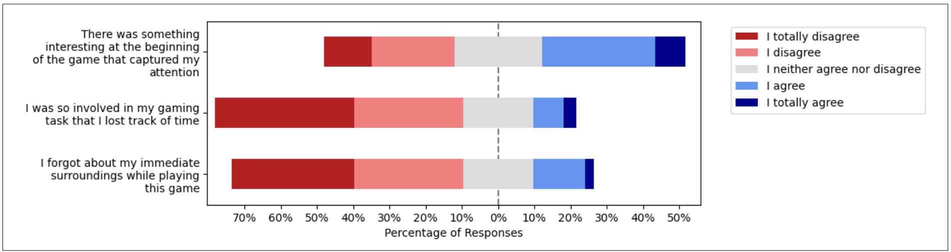


Fig. 17. Students' perceived focused attention.

(it could be faster) (PT-45)", and *"Usability, sometimes money and time took a long time to show any effect. I didn't know if I clicked or not (PT-56)"*.

It is important to note that the game is configurable, allowing the teacher to adjust the features to motivate students and help them maintain focus during gameplay. However, this hypothesis needs verification by evidence in future studies.

5.9. *Perceived Relevance*

The relevance factor, shown in Fig. 18, received positive ratings from students. The results show that most students (68%) agree that the educational offer of the game is consistent with their goals, in that they can relate the content covered to its real-world application in their future careers. Most of the participants highlighted the way the game explores the content as a positive point. They argued that has a preference for learning the content in this way. When participants were asked about their favorite aspect of the

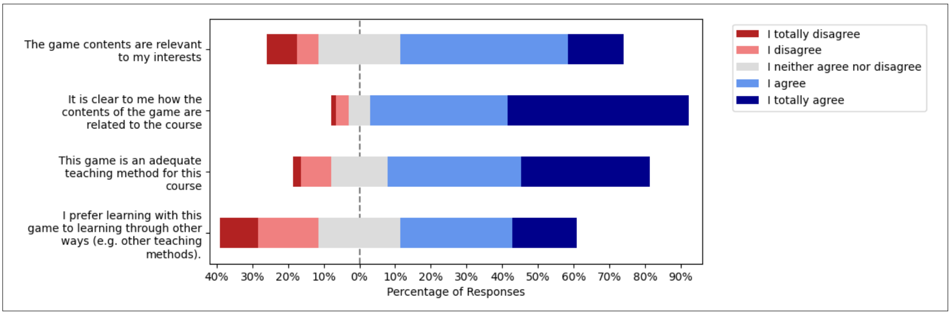


Fig. 18. Students' perceived relevance.

game, they mentioned: design and content *PT-20: "The look and application of the content of discipline in the rules of the game"*, gamification learning strategy *"The idea of gamify this content (PT-49)"*, *"The possibility of practicing what has been studied through a game (PT-52)"*, and *'What I liked most was being able to learn about MPS while playing, the division of steps and actions is interesting for learning (PT-57)"*.

5.10. *Perceived Learning*

The final factor evaluated was perceived learning. According to Fig. 19, the results for this factor were quite positive, more than 80% of the students agree that the game contributes, in general, to learning in the course, with approximately 75% of students agreeing that the game has a positive impact on their learning in the subject. Students reported that the game was more effective for learning than other activities offered in the course. Analysis of the results of the open questions reinforces this finding. When asked about the aspects they liked in the game, the participants mentioned: increasing knowledge *"Reinforce knowledge about MPS.BR (PT-56)"*, studying the topic more *"He encouraged me to study more about the MPS.BR process since I already had some related knowledge because I did a job on CMMI, which is one of the standards from which MPS.BR derives (PT-58)"*, and the notion of practice *PT-36: "You can get a certain notion of what should be done during a software development and where money should be invested during this process"*.

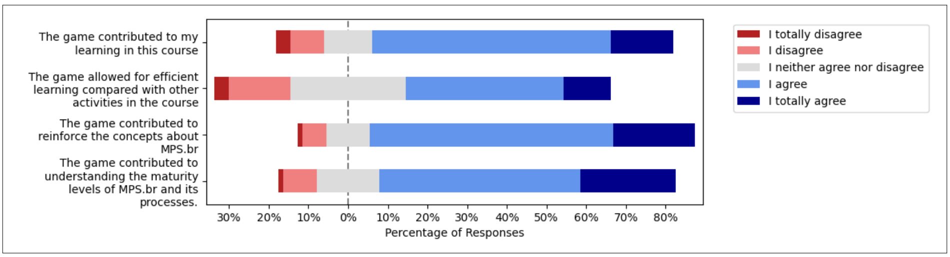


Fig. 19. Students' perceived learning.

In summary, the participants positively evaluated most dimensions, including usability, confidence, challenge, satisfaction, enjoyment, relevance, and perceived learning. This suggests that the *MPS Manager* game is effective and efficient for learning purposes.

The participants provided negative feedback on only three factors: social interaction, monotony related to enjoyment, and focused attention. The open questions revealed additional factors that may have contributed to these results, such as the lack of chat features for interacting with other participants and the time delay of the game. The following section will examine all the statements pointed by the participants as improvements for the game in the discursive answers. This analysis may aid in comprehending the survey results.

5.11. *Perceived Improvements for the MPS Manager*

This section presents the main points made by participants in response to an open question about improvements needed for the game. To clarify the discussion, certain statements have been highlighted.

The methodology employed for analyzing qualitative data obtained from open-ended questions involved coding. The process began with data collection, during which relevant information was gathered via a questionnaire featuring open-ended questions. Subsequently, the collected data were subjected to coding to identify recurring patterns and categorize them into distinct themes. These categories were then used to interpret the observations and provide a structured explanation of the data. All responses were thoroughly analyzed and classified according to the categories (or codes) identified during the coding process. This analysis was conducted manually by the research team, with content analysis employed to construct meaningful categories. Upon completing the analysis, the data were organized into five key categories: (I1) Usability – Aesthetics, (I2) Usability – Operability, (I3) Usability – Learnability, (I4) Social Interaction, and (I5) Challenge, each of which is discussed in detail below.

I1: Usability – Aesthetics*.* Some participants who answered the open question about the game's weaknesses mentioned problems related to usability design (28%) and response time (22%). Regarding design, participants revealed that *"Need to improve the graphical part and the front-end (PT-20)"*, and *"Improve the interactions because the buttons are slow to respond and some functions were not so clear (PT-8, PT39)"*. *PT-44* said: *"The game needs a more attractive UI"*. Also, *"The game needs to be optimized, offer better fluidity when showing the results (PT-4)"*.

I2: Usability – Operability*.* Improvements were also noted for operability factors (20%), such as difficulty in understanding level progression and game rules. For instance, *PT-28* explains: *"You could have scenery or stories between the years and feedback from the player's choices. In addition to improving the visibility of the level progression"*. According to the study's participants, *"The rules could be clearer, it could be clearer what was chosen, and it would be good to be able to go back before the year.*

It also requires a lot of clicking on the actions. It gets repetitive after a while (PT-30)". Additionally, *PT-64* said: *"The way the game works was a little confusing, making me miss several times because I didn't understand exactly what the purpose was"*.

I3: Usability – Learnability. Another point raised by the participants in their statements is the difficulty of learning to play the game. *PT-2* pointed out: *"Learning is not very clear and uninvited"*. As noted by *PT-17*: *"It is necessary instructions on how to play and achieve objectives"*.

I4: Perceived Social Interaction. Participants cited the inclusion of social interaction mechanisms between participants, such as chat, as another important improvement to the game. Participants indicated that: *"It is necessary to improve user interaction with the game (PT-60)"*. It was also suggested: *"creation of an interactive chat (PT-50)"*.

I5: Challenge Perception. Finally, in open-ended questions, participants reported the need to add more difficulty to the game to make it more challenging. *PT-46* stated that: *"It could add more difficulty to the game, for example, limiting the use of time and money to 'force' the player to make smart choices"*. Similarly, *PT-80* emphasizes that: *"needs to improve a sense of clearer progress, more challenging options, errors being pointed out (and perhaps punished)"*.

In summary, all the improvements suggested by the participants are relevant and reflect the need to optimize the game design, response time, and feedback progress. It is also necessary to present a complete description of the game's purpose and operation in the preparatory phase.

5.12. *Influence of Participant Profiles on Perceptions*

To assess how different participant profiles influenced students' perceptions of the *MPS Manager*, we analyzed responses across several demographic and contextual factors. Non-parametric tests were used due to the ordinal nature of the Likert-scale data. Specifically, we applied Mann-Whitney U tests for comparisons between two groups and Kruskal-Wallis tests for comparisons among three or more groups. We highlight below only the most relevant results connected to the learning objectives and engagement dimensions of the game.

5.12.1. *Academic and Institutional Factors*

Course Level (Basic vs. Advanced): Statistically significant differences were found in usability and social interaction. Advanced students reported higher agreement with usability aspects (83%) than basic students (70%) ($U = 412.5$, $p = 0.0401$). Conversely, basic students rated social interaction more positively. They reported 29% higher agreement with interaction features (72% vs. 43%) and 26% higher agreement with cooperation and competition aspects (89% vs. 63%) ($U = 334.5$, $p = 0.004$ and $U = 398.0$, $p = 0.022$).

Course Schedule (Day vs. Evening): Daytime students perceived the game as easier to use (9% higher agreement; $U = 558.0$, $p < 0.05$) and more socially engaging (22% higher agreement on interaction and cooperation; $p < 0.01$). In contrast, evening students reported 25% more focused attention, possibly due to lower peer interaction during gameplay.

Course Modality (Face-to-face vs. Remote): No statistically significant differences were found. However, remote students tended to report slightly higher agreement in challenge, focus, and confidence dimensions, potentially reflecting fewer classroom distractions.

University: Significant differences were found across institutions for enjoyment, relevance, and social interaction (Kruskal-Wallis, $p < 0.05$; Dunn's post-hoc with Bonferroni correction). UFGD students were consistently more positive, particularly in enjoyment (adjusted $p = 0.049$ compared to UNIFEI), challenge, and usability. UFF students showed higher satisfaction and learning perceptions, while UNIFEI students reported lower scores across most dimensions.

Didactic Support (Video vs. Lecture): Students who received live lectures instead of video lessons rated the game's visual design more positively ($M = 3.47$ vs. 2.88; $U = 1091.5$, $p = 0.013$). Although not statistically significant, they also reported lower perceived need for prior knowledge, possibly due to increased in-class interaction.

Game Training: Students who received prior training (UFF) agreed more strongly that most people would learn to play the game quickly ($U = 1077.5$, $p < 0.05$). Other differences across groups were minimal.

5.12.2. *Player Profiles*

Gaming Experience: No statistically significant differences were found across gamer experience levels. However, more experienced gamers reported higher enjoyment (73%), while beginner players perceived the game as more challenging and were less likely to recommend it. Despite this, 68% of all participants agreed on the relevance of the game, with even higher agreement among experienced gamers.

Gender: A statistically significant difference was found in rule clarity: men rated the rules clearer than women ($U = 202.0$, $p < 0.05$), which may be related to prior gaming exposure (66.7% of women reported low or no experience). For other dimensions, no significant gender differences were observed. It is also important to highlight that the sample size of the female audience is significantly smaller than that of the male audience, which may affect the representativeness and generalization of the results.

Age Group: Younger students (18–28) consistently reported higher usability scores than older students (29–39), including ease of play ($M = 3.49$ vs. 2.14; $U = 451.5$, $p = 0.018$), ease of learning ($U = 420.0$, $p = 0.009$), and rule clarity ($U = 396.0$, $p = 0.03$). These results likely reflect familiarity with digital environments. Furthermore, when asked whether most people would learn the game quickly, the younger group scored significantly higher ($M = 2.96$ vs. 1.86; $U = 405.0$, $p = 0.002$).

Table 5
Summary of significant findings by participant profile

Factor	Significant Findings
Course Level	Usability ↑ Advanced, Social ↑ Basic ($p < 0.05$)
Course Schedule	Usability, Social ↑ Day; Focus ↑ Evening ($p < 0.05$)
Course Modality	No significant differences (trends in focus, challenge)
University	Enjoyment ↑ UFGD vs UNIFEI ($p = 0.049$); Social, Relevance ($p < 0.05$)
Didactic Support	Design attractiveness ↑ Lecture group ($p = 0.013$)
Game Training	Ease of learning ↑ Trained group ($p < 0.05$)
Gaming Experience	No significant differences (trend: Enjoyment ↑ Experienced)
Gender	Rule clarity ↑ Male ($p < 0.05$)
Age Group	Usability ↑ Younger ($p < 0.05$)

Table 5 summarizes the most relevant findings by participant profile, highlighting statistically significant differences and observable trends across the evaluated dimensions.

These results reinforce the importance of considering learner characteristics when integrating serious games into educational contexts, particularly for teaching abstract and process-oriented topics like software quality models. The influence of academic level, institutional setting, and digital familiarity on perceived usability, social interaction, and engagement suggests that adapting instructional strategies and onboarding experiences may enhance the learning impact of serious games such as the *MPS Manager*.

6. Discussion

The results of this study provide evidence regarding the use of the *MPS Manager* as an educational resource to support the teaching of software quality improvement models. Concerning RQ1, most of the evaluated dimensions were positively rated by participants, particularly satisfaction, relevance, perceived learning, and usability – elements that are essential for the effectiveness of serious games in educational contexts. These findings suggest that the game contributes to students' understanding of the MPS.BR model by presenting process-related concepts through interactive and engaging simulations.

However, dimensions such as focused attention, rule clarity, and social interaction received lower ratings. These aspects appear to be critical to student engagement and the success of the learning experience: when game mechanics are not intuitive, objectives are unclear, or peer interaction is limited, the pedagogical potential of the tool may be undermined.

Regarding RQ2, the data show that individual and contextual characteristics significantly influenced students' perceptions. Factors such as age, prior gaming experience, course format, and institutional context directly impacted how participants understood, interacted with, and evaluated the *MPS Manager*. Younger students and those more familiar with digital games reported more positive experiences in usability and confidence, indicating that proficiency in game-based environments facilitates learning. Additionally, students who participated in face-to-face sessions – particularly those facilitated by instructors – reported higher satisfaction, more opportunities for collaboration, and a clearer understanding of the game's objectives. In contrast, students who accessed the game asynchronously, through video lessons, reported more difficulty in understanding the rules and mechanics, indicating that the learning context directly affects the effectiveness of the tool.

Based on these findings, several elements should be prioritized when adopting serious games to teach software quality models. The first relates to usability: the game should present clear rules, appropriate response time, and an intuitive interface. Many students reported difficulties interpreting their actions within the game, suggesting the need for improvements in feedback mechanisms, visual messaging, and interaction flow. A second critical aspect is onboarding. Students with limited familiarity with games or the *MPS-SW* model may benefit from interactive tutorials, introductory videos, or guided simulations before engaging with the main activity. This prior preparation may reduce the learning curve and enhance educational outcomes.

In addition, the absence of peer interaction mechanisms was identified as a limitation. Although the *MPS Manager* was designed as a single-player game, the data suggest that students desire collaborative and social dynamics, which may further enhance engagement. It is therefore recommended that future versions incorporate features such as group challenges, public rankings, peer chat, or cooperative gameplay mechanics. These elements could increase the game's motivational potential and make it more dynamic and appealing, especially for experienced players or those in classroom-based settings.

Beyond these points, qualitative feedback revealed that the perception of progression throughout the game was among the most valued aspects for students. Features such as level advancement, the ability to retry, and feedback linked to decision-making contributed to a sense of achievement – an important factor for motivation and learning. Moreover, many students reported that the learning provided by the *MPS Manager* was more effective than traditional instructional activities, such as readings or lectures. These findings suggest that serious games may be especially effective for teaching abstract or complex topics, such as software process improvement models.

The analysis also highlighted important institutional and contextual differences. Students enrolled in introductory courses, particularly those attending daytime in-person classes, reported higher engagement and perceived collaboration while using the game. These findings suggest that the effectiveness of the *MPS Manager* may be enhanced when applied in more interactive contexts, with greater instructional support and lower extraneous cognitive load – as is often the case in early-stage courses offered during daytime sessions.

This study also enables the formulation of practical recommendations for instructors who wish to adopt the *MPS Manager* in educational settings. We recommend that the game be implemented preferably in face-to-face or synchronous environments, with instructor support, allowing students to ask questions in real time and fostering peer interaction. The results suggest that the game is particularly effective in introductory courses and daytime classes, where engagement and collaboration are more easily fostered. It is also advisable that gameplay be preceded by a brief explanation of the *MPS-SW* model, along with clear instructions on the rules and objectives of the game. Furthermore, the game should provide continuous and visible feedback on student progress, emphasizing the sense of progression that emerged as a key motivator. Finally, since many students rated the *MPS Manager* as more effective than traditional instructional activities, its adoption may be planned as a complementary teaching strategy, enhancing the learning of abstract or challenging topics, such as software quality improvement models.

It is also worth noting that the *MPS Manager* is methodologically agnostic and can be integrated into courses that adopt different software development paradigms, such as agile, plan-driven, or hybrid models. This flexibility aligns with the *MPS-SW* model itself, which focuses on process quality rather than prescriptive development practices. Students' prior exposure to specific methodologies may influence how they approach decision-making within the game, offering educators an opportunity to draw connections between theoretical models and practical development experiences across diverse instructional contexts.

Although the game was configured using the *MPS-SW* model, it was developed with sufficient flexibility to support other quality improvement frameworks. Given that the underlying structure of the game relies on configurable levels, processes, and evaluation logic, it could be easily adapted to international models such as the Capability Maturity Model Integration (CMMI). This adaptability broadens the potential applicability of the *MPS Manager* beyond its initial context, allowing institutions to tailor the experience according to specific curricular or industry standards.

7. Conclusion

This paper presents the *MPS Manager*, a serious game based on the *MPS-SW* model aimed at providing students with an engaging experience to help them reach the highest maturity level within the *MPS-SW* model, thus facilitating learning and engaging students in topics related to software process improvement models. The game's goal is for the students to achieve the highest level of maturity within the *MPS-SW* model. The evaluation was conducted in three undergraduate courses at Brazilian universities, with a sample of 83 students. The results indicate that including the game in the classroom increased student engagement and satisfaction regarding software quality topics, enriching their learning experience.

The sample consisted of 83% male students between 19 and 29, with the majority having prior experience with digital games. The *MEEGA+* model (Petri *et al.*, 2017), which provides a methodology for evaluating educational games in the context of Computer Science education, was used as the basis for defining the questionnaire and analyzing responses. Student perceptions were analyzed across nine dimensions: usability, confidence, challenge, satisfaction, social interaction, enjoyment, focused attention, relevance, and perceived learning.

The results revealed predominantly positive evaluations in dimensions such as satisfaction, relevance, learning, and usability. Students highlighted the game's visual design, its applicability to the course content, and the sense of progress when overcoming the proposed challenges. Additionally, the analysis revealed perception differences related to student profiles, including age, gender, experience with digital games, and educational context (such as the course format and the use of resources like video lessons or in-person sessions).

Despite the promising results, some limitations of the *MPS Manager*'s evaluation must be acknowledged. The sample, consisting of 83 students, had a gender imbalance, with a predominance of male participants and those with prior experience in digital games. Future research could address this by evaluating with a more diverse and gender-balanced sample.

Another limitation is the absence of a direct quantitative assessment of the game's impact on students' academic performance. Although students' perceptions were positive regarding perceived learning, it would be interesting to correlate these perceptions with other data, such as grades in assessments, in future works to more accurately measure the game's effectiveness in promoting learning. Furthermore, while this study maintained an exploratory focus using non-parametric tests, future research could apply multivariate techniques – such as logistic regression or classification models – to better capture the combined influence of multiple variables on student engagement and learning outcomes.

The evaluation findings corroborate the student's acceptance of the *MPS Manager* and its capacity to facilitate learning by its stated objective. We plan to release this game on a broader platform called Quality Manager. Although still under development, the source code of the latest version is already available in an open-source repository[3]. Future improvements will continue to be shared through this repository.

Future work includes enhancing the *MPS Manager* based on student feedback, incorporating new features to assist students, and providing guidance on game progression. Further improvements will consist of creating more challenging activities and, most importantly, adding an integrated chat feature, which was identified as a priority by the study participants.

Acknowledgments

The authors gratefully acknowledge the support of the Brazilian National Council for Scientific and Technological Development (CNPq) (grant numbers: 440425/2024-7 and 420025/2023-5), the "Pró-Reitoria de Pesquisa e Inovação, Universidade de São Paulo" (PRPI-USP) (grant number: 22.1.09345.01.2), and PROPP/UFGD -SIGProj nº 322855.1174.8276.11032019.

[3] Quality Manager source code: `https://github.com/Quality-Gamer/qg-manager`

References

Alhammad, M.M., Moreno, A.M. (2018). Gamification in software engineering education: A systematic mapping. *Journal of Systems and Software*, 141, 131–150. https://doi.org/10.1016/j.jss.2018.03.065.

Alkassim, R.S., Tran, X., Rivera, J.D. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, 5(1), 1–4. https://doi.org/10.11648/j.ajtas.20160501.11

Aydan, U., Yilmaz, M., Clarke, P.M., O'Connor, R.V. (2017). Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach. *Computer Standards Interfaces*, 54, 129–138. Standards in Software Process Improvement and Capability Determination. https://doi.org/10.1016/j.csi.2016.11.014

Bai, S., Hew, K.F., Huang, B. (2020). Does gamification improve student learning outcome? Evidence from a meta-analysis and synthesis of qualitative data in educational contexts. *Educational Research Review*, 30, 100322. https://doi.org/10.1016/j.edurev.2020.100322 https://www.sciencedirect.com/science/article/pii/S1747938X19302908

Barreto, C.F., França, C. (2021). Gamification in Software Engineering: A literature Review. In: *2021IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pp. 105–108. https://doi.org/10.1109/CHASE52884.2021.00020

Calderón, A., Trinidad, M., Ruiz, M., O'Connor, R.V. (2019). An Experience of Use a Serious Game for Teaching Software Process Improvement. In: Walker, A., O'Connor, R.V., Messnarz, R. (Eds.), *Systems, Software and Services Process Improvement*. Springer International Publishing, Cham, pp. 249–259. https://doi.org/10.1007/978-3-030-28005-5_19

Chaves, R.O., von Wangenheim, C.G., Furtado, J.C.C., Oliveira, S.R.B., Santos, A., Favero, E.L. (2015). Experimental Evaluation of a Serious Game for Teaching Software Process Modeling. *IEEE Transactions on Education*, 58(4), 289–296. https://doi.org/10.1109/TE.2015.2411573

CMMI, I. (2021). CMMI Development v2.0. https://cmmiinstitute.com/cmmi

Cooper, K.M.L., Bucchiarone, A. (Eds.) (2023). *Software Engineering for Games in Serious Contexts -Theories, Methods, Tools, and Experiences*. Springer, Cham. https://doi.org/10.1007/ 978-3-031-33338-5

De Oliveira Colares, A.F., Furtado, J.C.C., Oliveira, S.R.B. (2023). Content and Skills for Teaching Software Process Improvement in the Computer Science Course: A Mapping of ACM / IEEE, SBC, SWEBOK, CMMI and MR-MPS-SW Assets. In: *2023 IEEE Frontiers in Education Conference (FIE)*, pp. 1–8. https://doi.org/10.1109/FIE58773.2023.10343447

de Sousa Borges, S., Durelli, V.H.S., Reis, H.M., Isotani, S. (2014). A systematic mapping on gamification applied to education. In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. SAC '14. Association for Computing Machinery, New York, NY, USA, pp. 216–222. https://doi.org/10.1145/2554850.2554956

Ekin, C.C., Polat, E., Hopcan, S. (2023). Drawing the big picture of games in education: A topic modeling-based review of past 55 years. *Computers Education*, 194, 104700. https://doi.org/10.1016/j.compedu.2022.104700. https://www.sciencedirect.com/science/article/pii/S0360131522002718

Furdu, I.M., Tomozei, C., Kose, U. (2017). Pros and cons gamification and gaming in classroom. *CoRR, abs/1708.09337*. http://arxiv.org/abs/1708.09337

Furtado, L.S., De Souza, R.F., Lima, J.L.D.R., Oliveira, S.R.B. (2021). Teaching Method for Software Measurement Process Based on Gamification or Serious Games: A Systematic Review of the Literature. *International JournalofComputerGamesTechnology*, 2021, 1–35. https://doi.org/10.1155/2021/8873997

Ghezzi, C., Mandrioli, D. (2005). The challenges of software engineering education. In: *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005.*, pp. 637–638. https://doi.org/ 10.1109/ICSE.2005.1553624

Gold-Veerkamp, C. (2021). Validated Undergraduates' Misconceptions about Software Engineering. In: *2021 IEEE Global Engineering Education Conference (EDUCON)*, pp. 609–618. https://doi.org/10.1109/EDUCON46332.2021.9454136

Holanda, M., Klysnney, T., Araujo, A., Silva, D.D., Oliveira, R.B., Koike, C.C., Castanho, C.D., Fachini-Gomes, J.B. (2022). Gender Diversity in STEM Graduate Programs at the University of Brasília in Brazil. In: *2022 IEEE Frontiers in Education Conference (FIE)*, pp. 1–9. https://doi.org/10.1109/FIE56618.2022.9962722

Kohwalter, T.C., Clua, E.W., Murta, L.G. (2011). SDM – An educational game for software engineering. In: *2011 Brazilian Symposium on Games and Digital Entertainment*, pp. 222–231. IEEE. https://doi.org/10.1109/SBGAMES.2011.10

Laine, T.H., Lindberg, R.S.N. (2020). Designing Engaging Games for Education: A Systematic Literature Review on Game Motivators and Design Principles. *IEEE Transactions on Learning Technologies*, 13(4), 804–821. https://doi.org/10.1109/TLT.2020.3018503

Likert, R.A. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 140(140), 44–53.

Marne, B., Wisdom, J., Huynh-Kim-Bang, B., Labat, J.-M. (2012). The Six Facets of Serious Game Design: A Methodology Enhanced by Our Design Pattern Library. In: Ravenscroft, A., Lindstaedt, S., Kloos, C.D., Hernández-Leo, D. (Eds.), *21st Century Learning for 21st Century Skills*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 208–221.

Martins, C., Giraffa, L.M.M., do Rosário Lima, V.M. (2018). Gamificação e seus potenciais como estratégia pedagógica no Ensino Superior. *RENOTE – Revista Novas Tecnologias na Educação*, 16(1). https://doi.org/10.22456/1679-1916.86005

Moura, V., Santos, G. (2018). ProcSoft: A Board Game to Teach Software Processes Based on ISO/IEC 29110 Standard. In: *Proceedings of the XVII Brazilian Symposium on Software Quality*. SBQS '18. Association for Computing Machinery, New York, NY, USA, pp. 363–372. https://doi.org/10.1145/3275245.3276319

Navarro, E.O., van der Hoek, A. (2004). SimSE: an educational simulation game for teaching the Software engineering process. In: *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. ITiCSE '04. Association for Computing Machinery, New York, NY, USA, p. 233. 1581138369. https://doi.org/10.1145/1007996.1008062

Ouhbi, S., Pombo, N. (2020). Software Engineering Education: Challenges and Perspectives. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*, pp. 202–209. https://doi.org/10.1109/EDUCON45650.2020.9125353

Petri, G., Gresse von Wangenheim, C., Borgatto, A.F. (2017). In: Lee, N. (Ed.) *MEEGA+, Systematic Model to Evaluate Educational Games*. Springer International Publishing, Cham, pp. 1–7. 978-3-319-08234-9. https://doi.org/10.1007/978-3-319-08234-9_214-1

Petri, G., von Wangenheim, C.G., A.F., A.F.B. (2019). Student Questionnaire. English version. MEEGA+ A model for evaluating educational games. http://www.gqs.ufsc.br/quality-evaluation/meega-plus/

Pinedo, L., Valles-Coral, M., Navarro-Cabrera, J.R., Injante, R., Cárdenas-García, , Ruiz-Saavedra, F., García-Rivas-Plata, C., Flores-Tananta, C.A. (2023). Software quality models: Exploratory review. *EAI Endorsed Transactions on Scalable Information Systems*, 10(6), 1–7.

Ribeiro, L., Barbosa, G., Silva, I., Coutinho, F., Santos, N. (2019). Um Panorama da Atuação da Mulher na Computação. In: *Anais do XIII Women in Information Technology*. SBC, Porto Alegre, RS, Brasil, pp. 1–10. https://doi.org/10.5753/wit.2019.6707

Shaw, M. (2000). Software engineering education: A roadmap. In: *Proceedings of the Conference on the Future of Software Engineering*, pp. 371–380.

Silva, A., Sousa, J., Vasconcelos, P., Rabelo, J. (2023). MPS.Br Game: Proposta de Jogo Educacional. *XIV Computer on the Beach*, 14, 495–497. https://doi.org/10.14210/cotb.v14.p495-497

SOFTEX (2021). Guia Geral MPS de Software. https://softex.br/download/guia-geral-de-software-2021/

Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education Limited, Boston.

Starr, C.W., Manaris, B., Stalvey, R.H. (2008). Bloom's taxonomy revisited: specifying assessable learning objectives in computer science. *SIGCSE Bull.*, 40(1), 261–265. https://doi.org/10.1145/1352322.1352227

Tonhão, S., Shigenaga, M., Herculani, J., Medeiros, A., Amaral, A., Silva, W., Colanzi, T., Steinmacher, I. (2023). Gamification in Software Engineering Education: A Tertiary Study. In: *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. SBES '23. Association for Computing Machinery, New York, NY, USA, pp. 358–367. https://doi.org/10.1145/3613372.3614193

I.R. Agualuza earned a Bachelor's degree in Computer Science from the Fluminense Federal University (UFF) in 2021. During his studies, he developed an interest in software engineering. He is currently working as a software developer in the industry, having contributed to projects in urban mobility and healthcare applications. Contact him at iagoagualuza@gmail.com.

S.M. Melo received the Ph.D. degree in Computer Science and Computational Mathematics at the Institute of Mathematics and Computer Science (ICMC) University of São Paulo (USP) in 2018. Actually is an assistant professor at the Faculty of Exact Sciences and Technology (FACET) Federal University of Grande Dourados (UFGD). Her research interests include computing education, software testing, and experimental software engineering. Contact her at silvanamelo@ufgd.edu.br.

L. Garcés is an assistant professor at the Institute of Mathematics and Computer Sciences (ICMC) of the University of São Paulo (USP). Her research interests include software engineering and its education, software architectures, software quality, IA and software engineering, and e-Health software systems. She received a Ph.D. in Computer Science from USP and the University of Southern Brittany (UBS), France, in 2018. She is a member of the academic societies ACM SIGSOFT and SBC. Contact her at linagarces@usp.br.

V.O. Neves received the Ph.D. degree in Computer Science from the University of São Paulo (USP) in 2015. She is currently an assistant professor at the Institute of Computing, Fluminense Federal University (UFF), Brazil. Her research interests include software testing and quality assurance, with a focus on testing complex systems, the intersection of artificial intelligence and software testing, and software engineering education. Contact her at vania@ic.uff.br.