Towards a Reference Architecture for Designing Open-Source Digital Games

Fernando Yang[1], Alessandreia Marta de Oliveira [2], Renan Vinicius Aranha [1], Pedro Henrique Dias Valle[1]

[1]University of São Paulo - São Paulo – SP – Brazil

[2]Universidade Federal de Juiz de Fora - Juiz de Fora – MG – Brazil

fernandoyang@ime.usp.br, alessandreia.oliveira@ufjf.br

{renanvinicius,pedrohenriquevalle}@usp.br

***Abstract. Introduction**: From the popularization of game engines, digital game development has become more accessible to both experts and non-experts. However, it remains complex, especially in open-source contexts, where collaboration and code evolution demand well-defined architectural structures. **Objective:** This work proposes a reference architecture, named OSGRA, to support open-source digital game development, addressing the need for organization, reuse, and quality in such environments. **Methodology**: To achieve this, the ProSA-RA process was adopted to construct the reference architecture. The proposed architecture was evaluated using the Technology Acceptance Model (TAM), considering quality requirements, as well as usability and acceptance. **Results:** The evaluation results indicate that the OSGRA satisfactorily meets the defined objectives, supporting key quality attributes such as interactivity and maintainability. Additionally, the results highlight opportunities for improvement that can guide future enhancements.*

***Keywords** Open-Source, Game Development, Quality Attributes, Reference Architecture, Software Architecture.*

1. Introduction

In recent years, the digital games industry has undergone significant expansion, reflected in the diversification of genres, platforms, and aesthetic experiences, which reinforces games as complex sociotechnical artifacts integrating narrative, interaction design, audiovisual elements, and software engineering [Valle et al. 2017, Kattilakoski 2019, Zhan et al. 2024, Lima et al. 2025]. However, game development remains inherently complex, requiring multidisciplinary knowledge [Genesio et al. 2023, Genesio et al. 2024, Salminen 2025]. In this context, open-source development emerges as a promising paradigm, fostering collaboration and transparency [Valle 2016, Scacchi 2004, Hieu 2025]. Nevertheless, its distributed nature introduces persistent challenges, including architectural inconsistencies, integration issues, and difficulties in maintaining quality, security, and long-term evolution [Steinmacher et al. 2015, Haider et al. 2023].

Software Architecture (SA) plays a fundamental role in addressing these challenges by structuring system components, relationships, and design decisions, enabling alignment between technical requirements and user experience goals

[Oliveira et al. 2022]. From this perspective, Reference Architectures (RAs) provide higher-level abstractions that consolidate domain knowledge, promote reuse, and guide the development of consistent and interoperable systems [Cloutier et al. 2010, Nakagawa et al. 2014].

Although RAs have been widely applied across domains [Roy et al. 2026, Bommakanti 2026, Pederson 2026], there is still a lack of consolidated solutions tailored to open-source digital games [Pascarella et al. 2018]. This gap forces developers to repeatedly design architectures from scratch, hindering reuse, consistency, and long-term sustainability [Folmer 2007, Pascarella et al. 2018, Mizutani et al. 2021]. To address this limitation, this work proposes a reference architecture for open-source digital games, named OSGRA, developed using the ProSA-RA process [Nakagawa et al. 2014]. The proposed architecture aims to support collaborative development by improving interoperability, facilitating maintenance and evolution, and establishing a shared design vocabulary. Furthermore, it is evaluated to ensure its practical applicability and alignment with the needs of the development community.

The remainder of this work is organized as follows. Section 2 presents the related work, discussing existing reference architectures in the domain of digital games and their limitations. Section 3 presents the research method used to conduct this work. Section 4 describes the establishment of the proposed OSGRA, detailing each step of the ProSA-RA process, including the investigation of information sources, definition of architectural requirements, and architectural design. Section 5 presents the evaluation of the architecture, along with the results and discussion. Section 6 discusses the threats to validity of this study. Finally, Section 7 concludes the work and outlines directions for future work.

2. Related Work

Although several studies have proposed RAs and models to support the development of complex software systems, particularly in domains such as serious games, digital games, and health platforms [Garcés et al. 2021], there are still few works specifically addressing the intersection of digital games and open-source development. In this context, this section briefly presents two representative works and discusses their limitations.

The work conducted by Carvalho (2017) introduces a comprehensive approach to serious game development by integrating a conceptual model and an RA. The Activity Theory-based Model for Serious Games (ATMSG) supports the analysis and design of educational games by structuring the interplay between gameplay, learning, and instructional activities. Additionally, the Service-Oriented Reference Architecture for Serious Games (SORASG) promotes reuse through service-based components, such as learning analytics and user profiling.

Mizutani and Kon (2020) proposed the Unlimited Rulebook (URB), a RA focused on economic mechanics in digital games. The architecture emphasizes modularity and flexibility by combining architectural styles, such as the Entity-Component-System (ECS) and the Adaptive Object Model (AOM), and techniques such as predicate dispatching. URB enables iterative development through dynamically loadable rule patches, supporting extensibility in complex game mechanics. Its evaluation, conducted through quasi-experiments with students, indicates that although it does not significantly simplify implementation, it reduces the effort required for extending mechanics.

Both SORASG and URB aim to improve reuse and reduce complexity in game development, but they differ in scope and completeness. SORASG provides a holistic approach by combining a conceptual model with a service-oriented architecture, explicitly addressing interoperability, quality attributes, and educational aspects. However, it is strongly tailored to serious games, limiting its generalizability. In contrast, URB focuses on flexibility and extensibility in its economic mechanics through modular and dynamic structures, but it lacks a broader architectural perspective, including explicit treatment of quality attributes, stakeholders, and integration with other subsystems.

The analyzed RAs do not consider the specific challenges of open-source game development, such as distributed collaboration and long-term evolution. This limitation reveals a gap and motivates the need for a new reference architecture that integrates modularity, reuse, and flexibility with explicit support for open-source practices.

Table 1. Comparative Analysis between SORASG and URB

Aspect	SORASG	URB
Scope	Full serious game architecture	Economic mechanics subsystem
Architectural Style	Service-Oriented Architecture (SOA)	ECS + AOM
Focus	Reuse, interoperability, and learning support	Flexibility and extensibility of mechanics
Conceptual Support	Yes (ATMSG model)	Not provided
Quality Attributes	Explicitly addressed	Not explicitly addressed
Evaluation	ATAM + Proof of Concept	Quasi-experiments
Domain Limitation	Restricted to serious games	Restricted to game economy

3. Research Method

This work adopts the ProSA-RA process [Nakagawa et al. 2014] to design the proposed RA (OSGRA), following four main steps shown in 1:

- **Step 1 – Investigation of Information Sources:** Relevant literature on RA and digital games was analyzed, including open-source systems and existing architectures. Additionally, we conducted a Systematic Mapping Study (SMS) to identify architectural solutions, technologies, and non-functional requirements;
- **Step 2 – Definition of Architectural Requirements:** We elicited system requirements from the identified sources and analyzed them to derive architectural requirements, considering open-source characteristics such as collaboration, modularity, extensibility, and transparency;
- **Step 3 – Reference Architecture Design:** We designed OSGRA based on the identified requirements, aiming to support reuse, collaboration, and key quality attributes (e.g., maintainability and scalability). The architecture was described using the 4+1 view model;
- **Step 4 – Evaluation of the Reference Architecture:** We evaluated the architecture using the Technology Acceptance Model (TAM), through a questionnaire assessing perceived usefulness and ease of use among experts and stakeholders.

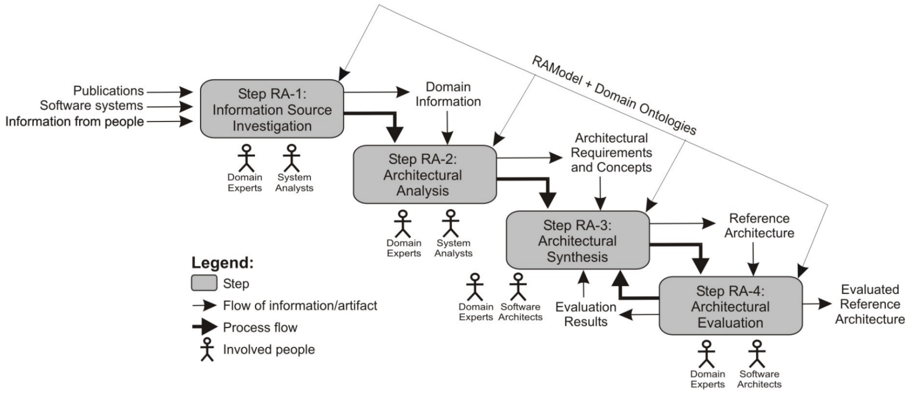

Figure 1. Overview of Research Method [Nakagawa et al. 2014].

4. Establishment of OSGRA

For the development of the RA for open-source digital games, named OSGRA (Reference Architecture for Open Source Games), the ProSA-RA process [Nakagawa et al. 2014] was adopted. ProSA-RA defines a systematic process composed of four main steps that guide the construction of RAs. We selected this process for its structured nature and its widespread adoption in the literature for building robust, well-founded RAs [Mizutani e Kon 2020, Michael e Shekhovtsov 2024, de Oliveira Camargo et al. 2024]. Each ProSA-RA step is described below, with specific attention to the context of open-source digital game development.

Step RA-1: Investigation of Information Sources

We considered information sources [Zimbrão et al. 2024], including RAs and related work in domains such as open-source systems and digital games, as well as architectures such as openKONSEQUENZ [Goering et al. 2017], SORASG [Carvalho 2017], URB [Mizutani e Kon 2020], and ePHoRt [Perez-Medina et al. 2019]. These studies provided insights into architectural solutions, design decisions, and quality concerns relevant to the development of open-source digital games.

Additionally, we conducted a Systematic Mapping Study (SMS) following a protocol based on the guidelines proposed by Kitchenham et al. (2015). The goal of this SMS was to identify and analyze existing studies on digital games, with a particular focus on architectural solutions, development processes, technologies, and non-functional requirements (NFRs). The SMS considered studies published up to February 2026 and employed a well-defined search string and study selection criteria across five databases (EI Compendex, IEEE Xplore, Web of Science, ScienceDirect, and Scopus) to enhance the coverage and completeness of studies related to the development of digital open-source games. As a result, a total of 19 primary studies from 1,137 candidates were analyzed, complemented by an analysis of existing open-source games, and the key characteristics, functionalities, architectural patterns, and requirements relevant to the domain were identified. These findings directly supported the construction of OSGRA by providing a consolidated view of the state of the art.

Step RA-2: Definition of Architectural Requirements

In the second stage of ProSA-RA, we used the information and domain concepts obtained from Step RA-1 to elicit requirements for the OSGRA design. As a result, a set of System Requirements (SRs) common to open-source digital games was identified, encompassing aspects of software engineering, open-source ecosystems, and digital game development.

It is important to note that some of these SRs were explicitly extracted from the analyzed studies, while others were identified implicitly through the analysis of existing architectures, platforms, and development practices in open-source digital games. This process also considered characteristics inherent to open-source projects, such as community collaboration, modularity, extensibility, and transparency.

Next, we conducted a detailed analysis of the identified requirements to derive the Architectural Requirements (ARs). This process involved qualitative analysis, consolidation of similar requirements, and discussions based on software architecture principles and quality attributes. As a result, we identified a set of key ARs that support the development of open-source digital games, as presented in Table 2.

Table 2. System Requirements and Architectural Requirements for OSGRA

ID	System Requirement	Architectural Requirement
AR-1	The game must support collaborative development among distributed contributors	The RA must provide mechanisms to support collaborative and distributed development, including version control integration and contribution workflows
AR-2	The game must be easily extensible with new features and content	The RA must adopt a modular and extensible architecture to allow the addition of new features and game content
AR-3	The game must ensure maintainability over time	The RA must promote maintainability through proper documentation, modularization, and separation of concerns
AR-4	The game must be adaptable to different platforms and environments	The RA must support portability, cross-platform deployment, and adaptability to varying system configurations
AR-5	The game must ensure good performance and responsiveness	The RA must include architectural mechanisms to ensure performance and low latency during gameplay
AR-6	The game must ensure functional correctness and appropriateness	The RA must provide mechanisms to guarantee that all game features are correctly implemented and that players can experience the intended content fully
AR-7	The game must ensure reliability, including fault tolerance and stability	The RA must incorporate strategies to prevent failures, ensure system stability, and minimize crashes or unexpected behavior during gameplay
AR-8	The game must provide transparency and accessibility of its source code	The RA must support open-source principles, ensuring code availability, traceability, and transparency
AR-9	The game must support user engagement and interaction	The RA must enable interactive mechanisms and feedback systems to enhance user engagement
AR-10	The game must support capacity scalability to handle varying numbers of concurrent users and content growth	The RA must be designed for efficient resource utilization and capacity management, ensuring performance stability under different load conditions

Step RA-3: Reference Architecture Design

Based on the elicited ARs, we designed OSGRA in Step RA-3. To properly describe the architecture, we defined the following key objectives: (i) to support the development process of open-source digital games, (ii) to promote the reuse of software components and assets, (iii) to facilitate collaboration among distributed contributors, and (iv) to improve key quality attributes such as maintainability, scalability, and extensibility.

Considering the ARs, we identified the main stakeholders involved in the open-source digital game ecosystem: players, developers, contributors, and organizations. Players interact with the games as end users, while developers and contributors collaboratively design, implement, and evolve the system. Organizations may adopt, support, or maintain open-source digital games in educational, research, or commercial contexts.

Based on the interests and concerns of these stakeholders, we defined architectural views following the "4+1" view model, which organizes the description of a SA using five concurrent views, each addressing specific concerns [Kruchten et al. 2006]. Accordingly, the adopted views include the Logical View, Process View, Physical View, Development View, and Use Case View. Due to space constraints, these views are not detailed here but can be accessed externally[1].

After analyzing evidence from studies [Haoues et al. 2017, Bi et al. 2018] and refining it through expert input, while also considering main quality attributes identified, we designed OSGRA using a Model-View-Controller (MVC) architecture for its strong support for separation of concerns and portability. This is combined with a Microkernel architecture to support distributed collaboration through its component-independent, plug-in-based nature, and with Entity-Component-System (ECS) principles to provide the performance, flexibility, and modularity required for games, enabling independent testing and evolution of game mechanics. Figure 2 presents an overview of OSGRA, highlighting its main components and their interactions.

On the **View** side, the architecture supports the execution of digital games across platforms, ensuring portability and accessibility for a diverse user base. The **Controller** is responsible for handling user input and interaction with the system. It includes multiple components, which capture user actions and translate them into commands that affect the game state.

The **Model** is the core of the architecture, encapsulating the game's main functionalities. It is organized into two main parts: the **Core** and the **Plug-in** modules. The Core includes fundamental components that execute game logic and manage system operations. The Plug-in module enables extensibility by allowing the integration of additional features through independent plug-ins (e.g., plug-in 1 to plug-in N), each containing its own systems and components.

In addition, the Logical View defines the main components, responsibilities, and relationships, making the adoption of ECS and MVC explicit. The Process View describes runtime behavior through sequence diagrams. The Development View organizes the architecture into modules and shows how the Microkernel architecture supports

[1] Available at: `https://tinyurl.com/mwme8zdn`

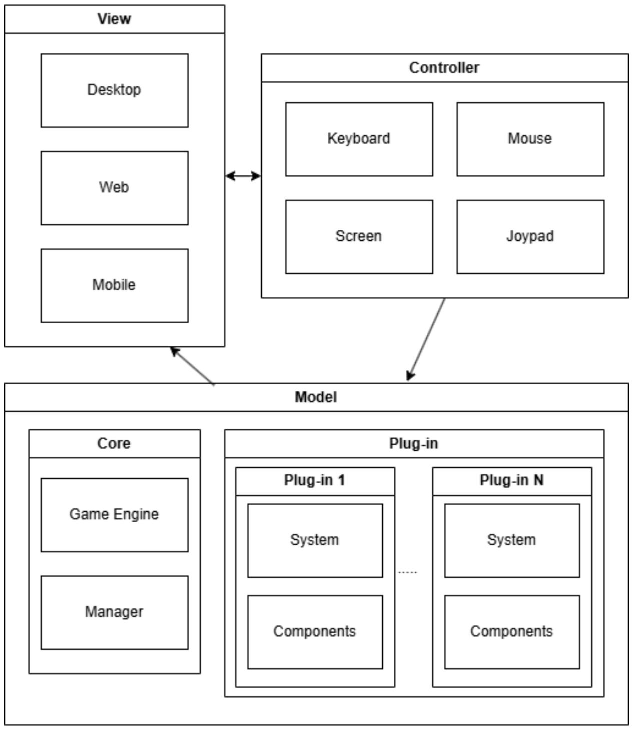

Figure 2. Overview of OSGRA

independently developed extensions. The Physical View specifies deployment structures, including a optional server services, demonstrating portability across environments. Finally, the Use Case View validates the architecture against stakeholder concerns, particularly those related to open-source development. This modular organization promotes separation of concerns, facilitates maintenance, and supports system evolution by enabling the addition of new functionality without affecting the game's core structure.

Step RA-4: Evaluation of the Reference Architecture

We evaluated OSGRA using the Technology Acceptance Model (TAM) to analyze the perceived usefulness, whether OSGRA effectively supports the development of open-source digital games, improves productivity, and contributes to software quality, and perceived ease of use, how easily the architecture can be understood, learned, and applied by developers and contributors, from the perspective of potential users [Davis et al. 1989]. TAM is a widely adopted model for assessing user acceptance of technologies, focusing on how users perceive the benefits and usability of a given solution. Additionally, we employed the Architecture Tradeoff Analysis Method (ATAM), which aims to assess whether OSGRA meets the identified quality requirements and to evaluate trade-off points between these attributes.

The evaluation involved three experts from areas related to the research field, identified as PA1, PA2, and PA3, as well as eight stakeholders relevant to the use of the system, identified as S1 to S8, totaling 11 participants with a background in Software Engineering and experience in digital games or software architecture. Their academic qualifications included three doctorates, one master's degree, and seven bachelor's degrees. The participants were selected through convenience sampling based on their familiarity with the domain and their ability to assess architectural solutions. Their experience ranged from beginner to advanced in quality attributes, open source, and game development, and most demonstrated moderate or advanced knowledge in at least two of these areas. The invitations were sent via email, and all participants were informed of the study's objectives, with participation being voluntary.

The evaluation process was structured in three main steps. First, participants received a document describing OSGRA, including its objectives, stakeholders, and architectural views. Next, a questionnaire was administered to collect demographic and professional information. Finally, participants completed the TAM-based questionnaire, composed of statements on perceived usefulness (PU) and perceived ease of use (PEOU), and an ATAM-based questionnaire, evaluating whether OSGRA is adequate for its purpose and meets the quality attributes; both instruments used a Likert scale and provided space for qualitative responses.

To support the evaluation process, a communication channel was made available to clarify participants' questions and doubts. The results from the questionnaires provide insights into the architecture's acceptance, indicating its potential applicability and usefulness in open-source digital game development.

5. Results and Discussion

This section mainly discusses the results of the evaluation of the proposed OSGRA using TAM, a widely used theoretical model for assessing user acceptance of technologies, based on two main constructs: Perceived Usefulness (PU) and Perceived Ease of Use (PEOU). PU refers to the degree to which a user believes that a system improves their performance, while PEOU refers to how effortless the system is perceived to be.

The evaluation was conducted with eleven participants (N = 11) with experience in Software Engineering and digital game development. The TAM questionnaire consisted of 12 items: 6 related to PU and 6 related to PEOU, evaluated on a 5-point Likert scale. Table 3 presents the questionnaire used in this study.

Table 3. TAM questionnaire for OSGRA evaluation

ID	Construct	Statement
P1	PU	The architecture would allow me to develop games more quickly.
P2	PU	The architecture would improve the quality of developed games.
P3	PU	The architecture would increase my development productivity.
P4	PU	The architecture would increase my effectiveness in game development.
P5	PU	The architecture would make it easier to develop games.
P6	PU	I would consider this architecture useful for game development.
P7	PEOU	Learning to use this architecture would be easy for me.
P8	PEOU	It would be easy to make the architecture do what I want.
P9	PEOU	My interaction with the architecture would be clear and understandable.
P10	PEOU	I would consider this architecture flexible to interact with.
P11	PEOU	It would be easy for me to become skillful in using this architecture.
P12	PEOU	I would consider this architecture easy to use.

The results obtained from the TAM evaluation are summarized in Table 4. The results indicate a positive perception of the architecture. The average score for Perceived Usefulness (PU) was 4.27, while Perceived Ease of Use (PEOU) reached 4.28, suggesting that participants consider the architecture both useful and relatively easy to use.

Table 4. TAM results provided by the evaluators

ID	PA1	PA2	PA3	S1	S2	S3	S4	S5	S6	S7	S8	Mean
P1	4	3	4	5	5	1	5	5	2	4	4	3.82
P2	5	5	5	5	5	4	5	5	3	5	4	4.64
P3	5	4	4	5	5	2	5	5	4	4	4	4.27
P4	4	5	4	5	5	2	5	5	4	4	4	4.27
P5	5	4	5	5	5	1	4	5	2	4	4	4.00
P6	4	5	5	5	5	3	5	5	5	5	4	4.64
P7	5	5	4	5	5	1	4	4	2	4	4	3.90
P8	2	5	5	5	5	3	5	5	4	4	5	4.36
P9	5	5	5	5	5	3	4	5	3	4	5	4.45
P10	5	5	5	5	5	5	4	4	4	5	5	4.72
P11	4	5	5	4	5	3	3	5	2	5	5	4.18
P12	5	5	5	5	5	1	4	5	2	4	4	4.09

Regarding PU, the highest scores were observed in P2 and P6 (4.64), indicating strong agreement that the architecture improves game quality and is useful for development. Lower scores in P1 (3.82) and P5 (4.00) suggest that participants are more cautious about short-term productivity gains. Qualitative feedback indicates that these limitations are mainly related to the learning curve and the need for comprehensive documentation. Participants emphasized that the architecture's benefits are more evident in long-term activities, such as maintenance and evolution.

Regarding PEOU, the highest score was observed in P10 (4.72), highlighting the architecture's flexibility. However, lower scores in P7 (3.90), P11 (4.18), and P12 (4.09) indicate challenges related to learning and mastering the architecture. These challenges are mainly associated with its structural complexity. The use of the MVC pattern was identified as a positive factor, contributing to the separation of concerns and improving conceptual understanding. However, participants highlighted that the effectiveness of this design depends on clear documentation and practical examples. Without such support, the architecture may impose a cognitive overhead, especially for less experienced developers.

As support for these arguments, the ATAM evaluation reveals that the architectural complexity of OSGRA can compromise the effectiveness of the quality attributes. Furthermore, in its open-source context, inadequate documentation and the complexity of understanding the architecture for new contributors pose significant risks to community engagement.

Overall, the results reveal a trade-off between architectural robustness and ease of adoption. While the architecture supports important quality attributes such as maintainability and extensibility, its complexity may hinder adoption in smaller projects. Finally, it is important to note that the results are based on a limited sample (N = 11). Future work should include evaluations with a larger, more diverse group of participants, as well as empirical studies in real-world settings.

6. Threats to Validity

This work collected empirically grounded evidence on OSGRA and used TAM and ATAM methods for its evaluation. As with any empirical study, this approach is subject to potential threats to validity. These threats are discussed according to the classification proposed by Wohlin *et al.* (2012).

Regarding **internal validity**, the construction of OSGRA may have been influenced by the authors' interpretations during the identification and transformation of system requirements into architectural requirements. To mitigate this threat, we followed a systematic process (ProSA-RA), which structures the stages of investigation, requirement elicitation, and architectural design. Additionally, the use of systematic methods helped reduce subjective bias during evaluation by employing standardized constructs.

Concerning **external validity**, the number of participants (N = 11) in the evaluation may limit the generalization of the results. However, to mitigate this threat, participants were selected based on their background in Software Engineering and familiarity with software architecture and digital games. Their experience provided valuable insights into the architecture's applicability and usefulness.

Finally, regarding **construct validity**, the use of TAM and ATAM as evaluation instruments may not capture all aspects of an architecture. However, we chose these artifacts due to their widespread use and their ability to assess user acceptance - focusing on perceived usefulness and ease of use - and to evaluate quality attributes, which are important indicators of the architecture's potential adoption.

7. Conclusion and Future Work

This work presented the development and evaluation of OSGRA, an RA designed to support the development of open-source digital games. Based on the TAM and ATAM evaluation results, participants perceived the architecture as useful and relatively easy to use, indicating its potential to support the development process and improve the quality of digital games.

The results indicate that OSGRA offers a solid architectural foundation for digital game development, as its design promotes modularity and extensibility, and the adoption of the MVC pattern effectively separates concerns, facilitating maintenance and evolution; nevertheless, while the proposed RA demonstrates promising acceptance and applicability, its usability and comprehensibility are hindered by insufficient documentation clarity and a lack of detailed guidance for architectural instantiation, which must be addressed to ensure effective real-world adoption.

As future work, we intend to refine the architectural documentation, provide practical examples of instantiation, and conduct additional evaluations with a larger and more diverse group of participants. We also plan to explore the application of OSGRA in real-world development projects, aiming to further validate its usefulness and effectiveness in supporting the development of open-source digital games.

Use of Artificial Intelligence

Generative Artificial Intelligence tools were used only for language review and minor editorial support. ChatGPT and DeepSeek were used for text refinement, and Grammarly for minor proofreading. All methodological decisions, analyses, and conclusions remain the sole responsibility of the authors.

References

Bi, T., Liang, P., e Tang, A. (2018). Architecture patterns, quality attributes, and design contexts: How developers design with them. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pages 49–58. IEEE.

Bommakanti, S. K. (2026). Cloud-native distributed ai: Enabling secure collaborative learning on heterogeneous infrastructure. *International Journal of Intelligent Systems and Data Science*, 1(1).

Carvalho, M. B. (2017). Serious games for learning: a model and a reference architecture for efficient game development.

Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., e Bone, M. (2010). The concept of reference architectures. *Systems Engineering*, 13(1):14–27.

Davis, F. D., Bagozzi, R. P., e Warshaw, P. R. (1989). Technology acceptance model. *J Manag Sci*, 35(8):982–1003.

de Oliveira Camargo, M. P., dos Santos Pereira, G., Almeida, D., Bento, L. A., Dorante, W. F., e Affonso, F. J. (2024). Ra4self-cps: a reference architecture for self-adaptive cyber-physical systems. *IEEE Latin America Transactions*, 22(2):113–125.

Folmer, E. (2007). Component based game development–a solution to escalating costs and expanding deadlines? In *International symposium on component-based software engineering*, pages 66–73. Springer.

Garcés, L., Martínez-Fernández, S., Oliveira, L., Valle, P., Ayala, C., Franch, X., e Nakagawa, E. Y. (2021). Three decades of software reference architectures: A systematic mapping study. *Journal of Systems and Software*, 179:111004.

Genesio, N. O. S., de Menezes, M. C. R., de Almeida, J. V. C., Boaventura, A. P. F. V., e Valle, P. H. D. (2023). Aprendendo lógica de programaçao desenvolvendo jogos digitais: Um relato de experiência. In *Workshop de Informática na Escola (WIE)*, pages 375–386. SBC.

Genesio, N. O. S., de Oliveira, A. M., Oliveira, E. W., e Valle, P. H. D. (2024). Panorama de estudos sobre jogos educacionais digitais em educação em computação. In *Workshop sobre Educação em Computação (WEI)*, pages 737–749. SBC.

Goering, A., Meister, J., Lehnhoff, S., Herdt, P., Jung, M., e Rohr, M. (2017). Reference architecture for open, maintainable and secure software for the operation of energy networks. *CIRED 24*, 2017(1):1410–1413.

Haider, S., Khalil, W., Al-Shamayleh, A. S., Akhunzada, A., e Gani, A. (2023). Risk factors and practices for the development of open source software from developers' perspective. *IEEE Access*, 11:63333–63350.

Haoues, M., Sellami, A., Ben-Abdallah, H., e Cheikhi, L. (2017). A guideline for software architecture selection based on iso 25010 quality related characteristics. *International Journal of System Assurance Engineering and Management*, 8(Suppl 2):886–909.

Hieu, H. V. (2025). Open-source software development and community management.

Kattilakoski, J. (2019). The growth of video game industry: Current status and future prospects.

Kitchenham, B. A., Budgen, D., e Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*. CRC press.

Kruchten, P., Obbink, H., e Stafford, J. (2006). The past, present, and future for software architecture. *IEEE software*, 23(2):22–30.

Lima, L., Costa, I., Bonzinho, C., Correia, S., e Martins, C. (2025). Mnema: Bridging research and art to combat gender inequity in the gaming sector. In *International Workshop on Interaction and Player Research in Game Development*, pages 67–83. Springer.

Michael, J. e Shekhovtsov, V. A. (2024). A model-based reference architecture for complex assistive systems and its application. *Software & Systems Modeling*, 23(5).

Mizutani, W. K., Daros, V. K., e Kon, F. (2021). Software architecture for digital game mechanics: A systematic literature review. *Entertainment Computing*, 38:100421.

Mizutani, W. K. e Kon, F. (2020). Unlimited rulebook: A reference architecture for economy mechanics in digital games. In *2020 IEEE International Conference on Software Architecture (ICSA)*, pages 58–68. IEEE.

Nakagawa, E. Y., Guessi, M., Maldonado, J. C., Feitosa, D., e Oquendo, F. (2014). Consolidating a process for the design, representation, and evaluation of reference architectures. In *2014 IEEE/IFIP Conference on Software Architecture*, pages 143–152. IEEE.

Oliveira, B. R., Garcés, L., Lyra, K. T., Santos, D. S., Isotani, S., e Nakagawa, E. Y. (2022). An overview of software architecture education. In *Congresso Ibero-Americano em Engenharia de Software (CIbSE)*, pages 76–90. SBC.

Pascarella, L., Palomba, F., Di Penta, M., e Bacchelli, A. (2018). How is video game development different from software development in open source? In *Proceedings of the 15th International Conference on Mining Software Repositories*, pages 392–402.

Pederson, J. (2026). Architecting trust: A modular framework for the operational deployment of autonomous systems.

Perez-Medina, J.-L., Jimenes-Vargas, K. B., Leconte, L., Villarreal, S., Rybarczyk, Y., e Vanderdonckt, J. (2019). ephort: towards a reference architecture for tele-rehabilitation systems. *IEEE Access*, 7:97159–97176.

Roy, J., Singh, S. K., e Shaw, L. (2026). Customer churn prediction on structured data using ft-transformer and stacking ensembles. *IEEE Access*.

Salminen, L. (2025). Managing multidisciplinary video game development projects and multidisciplinary project teams.

Scacchi, W. (2004). Free and open source development practices in the game community. *IEEE software*, 21(1):59–66.

Steinmacher, I., Silva, M. A. G., Gerosa, M. A., e Redmiles, D. F. (2015). A systematic literature review on the barriers faced by newcomers to open source software projects. *Information and Software Technology*, 59:67–85.

Valle, P. H. D. (2016). *Jogos educacionais: uma contribuição para o ensino de teste de software*. PhD thesis, Universidade de São Paulo.

Valle, P. H. D., Rocha, R. V., e Maldonado, J. C. (2017). Testing game: An educational game to support software testing education. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering*, pages 289–298.

Zhan, Y., Wang, T., e Bi, X. (2024). Creative production in the digital age: A network analysis of the digital game industry in china. *Geoforum*, 157:104158.

Zimbrão, R. D. O., Oliveira, A. M. D., Oliveira, E. W., e Valle, P. H. D. (2024). Essential aspects in digital educational games to support the teaching-learning of computing contents. In *Proceedings of the 20th Brazilian Symposium on Information Systems*, pages 1–10.