Framing Cloud Migration of Legacy Systems as Technical Sustainability Requirements within a Proprietary Software Ecosystem

Claudio Coelho da Silva Junior[1][0009−0001−0164−9835], Juliana Carvalho Silva do Outão[1][0000−0002−8630−1386], Luiz Alexandre Martins da Costa[1][0000−0002−8761−2135], Awdren de Lima Fontão[2][0000−0002−2988−9646], and Rodrigo Pereira dos Santos[1][0000−0003−4749−2551]

[1] Federal University of the State of Rio de Janeiro, Brazil
[2] Federal University of Mato Grosso do Sul, Brazil
{claudiocoelho, juliana.carvalho,luiz.costa}@edu.unirio.br, rps@uniriotec.br, awdren.fontao@ufms.br

Abstract. The increasing complexity of software ecosystems (SECO) demands effective governance mechanisms to ensure long-term sustainability, particularly within proprietary SECO (PSECO). As organizations modernize their technology platforms, cloud migration emerges as a key strategy to improve scalability, flexibility, and operational resilience. However, selecting an appropriate migration approach for legacy software assets requires a structured decision-making process that aligns with governance principles, business continuity, and technical sustainability. This work investigates cloud migration strategies within a PSECO and proposes a decision tree artifact to assist IT managers in evaluating migration options. We analyze real-world constraints, stakeholder concerns, and governance requirements through a participative case study in a large international organization. Our findings revealed that migration decisions are influenced not only by technical factors, such as architecture and performance, but also by business drivers, regulatory constraints, and organizational culture. The proposed guide provides a systematic approach to balancing modernization efforts with risk mitigation, helping organizations avoid technical debt while adapting to evolving industry demands. Using governance mechanisms in cloud migration strategies may support organizations in maintaining platform stability while fostering continuous innovation. Future research should explore how emerging cloud technologies and governance frameworks further impact modernization in the PSECO context.

Keywords: Software Ecosystems · Software Asset · Cloud Migration · Sustainability · Legacy Systems · Requirements

1 Introduction

The rise of software ecosystems (SECO) has transformed how organizations develop and manage software, shifting from isolated applications to collaborative, interlinked networks [21]. According to Manikas and Hansen [20], SECO consists of a dynamic set of actors within a shared platform, comprising software artifacts and technologies that enable the creation of diverse solutions and services.

As a subset, proprietary SECO (PSECO) are characterized by the centralization of intellectual property and governance under a leading organization, the keystone. This organization manages contributions, security, and compliance via confidentiality agreements [19]. A keystone defines governance frameworks to ensure long-term PSECO sustainability. Effective governance is essential to preserve a stable and adaptable platform, resistant to obsolescence and business changes [2,12]. As organizations invest in innovation and software assets to foster growth, a cycle of updates and implementations is needed [26]. However, evolving without compromising stability and availability remains challenging.

A key to modernizing without affecting stability and availability is technical sustainability, which ensures that assets remain maintainable, scalable, and aligned with business goals over time [29]. According to Lago et al. [17], software sustainability involves four dimensions: technical, economic, social, and environmental. In SECO, technical sustainability is tied to continuous platform evolution and avoiding obsolescence. Poor modernization leads to technical debt, higher maintenance costs, and integration issues with new technologies [18].

One strategy to ensure technical sustainability is cloud migration [3]. Assunção et al. [4] identify "cloudification" as a key approach for modernizing legacy systems, offering scalability, flexibility, and cost efficiency. From a sustainability view, cloud migration extends the software lifecycle, mitigating obsolescence while enhancing interoperability and automation.

A cloud strategy should embed governance mechanisms that enable transparency in decisions, infrastructure usage, and performance, building trust among ecosystem actors [28]. Migration must address legacy systems, rigid architectures, regulations, and change resistance. Including governance and metrics helps manage risks and ensure stability [24].

This study investigates how cloud migration strategies for legacy software assets can be expressed as a technical sustainability goal in a PSECO. To address this challenge, our study aims to develop a guide to support IT managers in selecting cloud migration strategies for legacy systems within a platform, aligning such decisions with principles of technical sustainability. Through a participative case study in a large international organization, we engaged stakeholders to explore real migration challenges. The resulting guide presents a structured approach to help IT teams select appropriate strategies aligned with governance, business continuity, and long-term sustainability.

2 Background and Related Work

SECO consists of a common technological platform, internal and external developers, and experts serving a user community to build valuable solutions. SECO can be classified into proprietary, open source, and hybrid based on the nature of source code protection and revenue generation models [20]. Our work focuses on PSECO, where software artifacts are protected by confidentiality agreements.

The governance of software assets within a technological platform is not limited to business and legal considerations. It is also essential for ensuring the long-term sustainability of the ecosystem [2]. As software assets evolve, maintaining technical sustainability becomes necessary to prevent obsolescence and ensure continuous alignment with business needs.

Sustainability, as defined by Hilty et al. [13], refers to the ability to preserve a system's functionality over time. Lago et al. [17] further emphasize that sustainability in software systems should be analyzed across four dimensions: economic, social, environmental, and technical. Among these, technical sustainability is particularly relevant for SECO governance, as it ensures that software remains maintainable, adaptable, and resilient, supporting the continuous evolution of the ecosystem.

Zacarias et al. [33] describe technical sustainability as the ability of SECO to evolve with technological changes while remaining functional and relevant. In PSECO, one of the challenges is deciding when and how to modernize software assets to maintain their longevity. A widely used approach is cloud migration [4], which enhances scalability, efficiency, and resilience.

The cloud migration strategy guides the transition from on-premises infrastructure to a cloud-based environment [14]. Since not all assets require migration, identifying which ones to move and aligning them with business and technical goals is crucial. Among the several cloud migration strategies proposed in the literature, the "7Rs" of cloud migration by Amazon Web Services (AWS) is widely recognized for its structured and practical approach [1,5,23]:

1. **Rehost (also known as "lift-and-shift")**: Move the software asset to the cloud without modifying its code or architecture. It is a quick, low-effort option to reduce operational costs efficiently;
2. **Replatform**: Make small infrastructure or configuration changes to optimize the software asset's performance in the cloud without significantly altering its architecture or code;
3. **Refactor**: Modify parts of the software asset's code to improve its compatibility with cloud-native features, such as containerization and managed services, while preserving its core architecture;
4. **Rearchitect**: Redesign the the software asset's architecture to take full advantage of cloud-native capabilities, such as microservices and serverless computing, often requiring significant code restructuring;
5. **Repurchase**: Replace the existing software asset with an equivalent SaaS (Software as a Service) solution, eliminating the need for infrastructure management and maintenance;
6. **Retire**: Identify the software assets that are no longer needed and decommission them, reducing costs and complexity; and
7. **Retain**: Keep certain the software assets on-premises due to technical, regulatory, or strategic reasons.

As related work, cloud migration has been studied in recent years. Moreover, several research efforts have proposed tools, frameworks, strategies, and methodologies to support organizations in transitioning their software assets from on-premises architectures to cloud-based environments.

Shastry et al. [23] evaluated cloud migration strategies for legacy systems, focusing on their adaptation to cloud environments. The study applied three key approaches: rehost, replatform, and rearchitect. Chinamanagonda [5] highlighted automation and cloud-native tools as key to efficient and error-free migration. Ahmad et al. [1] examined cloud migration frameworks and industry practices from major providers such as AWS, Azure, and Oracle Cloud.

Unlike related work, we focus on proposing a requirements-driven approach to defining migration strategies. We developed a decision tree artifact to instantiate the cloud migration strategy guide. This artifact was also validated in collaboration with industry practitioners through a participative case study.

3 Research Method

This work aims to develop a guide to support IT managers in decision-making regarding the choice of a cloud migration strategy for the legacy software assets of a technological platform as an attribute of technical sustainability in a PSECO. The RQ is: *"How can cloud migration strategy to legacy software assets be expressed as a technical sustainability goal in a PSECO?"*

To answer the RQ, we grounded our work in methods from the Empirical Software Engineering (ESE) guidelines [31]. Thus, to develop a cloud migration strategy guide proposal, we employed a research method comprising two phases: (I) Exploratory Phase; and (II) Design Phase. Fig. 1 illustrates the steps included in each phase of the research method. The Exploratory Phase involves empirical studies that identify governance challenges and migration patterns, while the Design Phase translates these findings into requirements and the construction of a decision-making guide. Each of these steps is further detailed in the following subsections. Our approach was inspired by the studies of Zacarias et al. [32] and Santos [21], who similarly derived the requirements for a solution based on previous research.

3.1 Exploratory Phase

This phase encompassed five studies, labeled S1 through S5, conducted between 2020 and 2024, investigating governance mechanisms in PSECO. The studies S1, S3, S4, and S5 were exploratory, employing empirical methods such as case studies, experiments, and data analysis, while S2 was a longitudinal literature study (LLS) that refined previous research on governance and health indicators.

These studies provided empirical evidence on governance challenges and strategies in PSECO, highlighting emerging guidelines on cloud migration strategies for legacy software assets. Tab. 1 presents a brief description that contains the objective, research method, and results of these studies. We also add the year they were carried out.

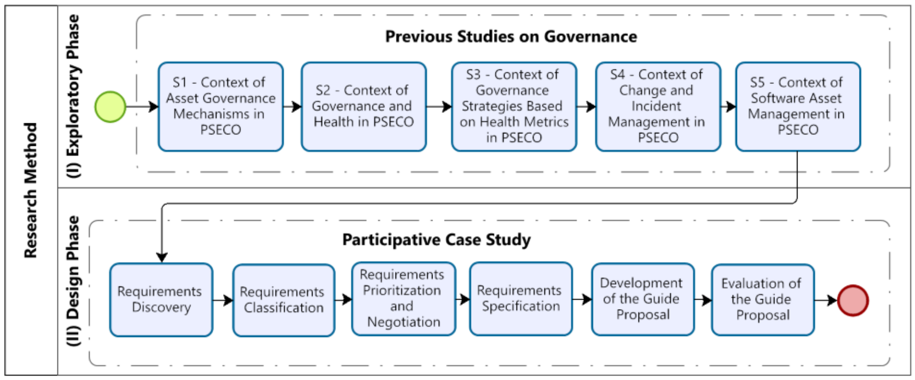

Fig. 1. Research method inspired by Zacarias et al. [32] and Santos [21].

3.2 Design Phase

Based on the findings of previous studies, our research group identified challenges in planning and conducting assessments to determine the best strategy for migrating software assets to the cloud. Each study required a distinct approach to evaluating technical, economic, and operational factors, taking into account variations in system architecture, dependency constraints, compliance requirements, and organizational goals.

This complexity highlighted the need for a structured approach with guidelines, tools, and artifacts to assist IT management in selecting the most suitable cloud migration strategy. Our guide supports IT managers in defining migration strategies by addressing migration patterns, cost-benefit analysis, and governance policies. To explore both the technological and social dimensions, we adopted a participative case study as research method. This approach enhances understanding of the investigated scenarios and uncovers new relationships, as inspired by the approach proposed by Trinkenreich et al. [27].

Together with practitioners, we developed a technically robust and operationally relevant cloud migration guide, tailored to the specific challenges of IT management in a global organization within a PSECO. To address this need, we initiated a requirement elicitation process to develop a guide to define strategies for migrating software assets to the cloud in PSECO. This phase consisted of five steps: (i) Requirements Discovery; (ii) Requirements Classification; (iii) Requirements Prioritization and Negotiation; (iv) Requirements Specification; (v) Development of the Guide Proposal; and (vi) Evaluation of the Guide Proposal. The first four stages were inspired by the requirements elicitation and analysis process proposed by Sommerville [25].

(i) Requirements Discovery: We applied document analysis as described by Wiegers and Beatty [30] to understand existing systems and define preliminary requirements, minimizing elicitation time. The documents analyzed are summarized in Tab. 1. Using open coding from the Grounded Theory [6], we

Table 1. Governance studies in PSECO used for requirements elicitation.

ID	Description	Year
S1	The study examined governance mechanisms in a PSECO and their impact on software asset management. Surveys and interviews highlighted innovation promotion, effective communication, and knowledge sharing as key factors, along with challenges such as resistance to change [7].	2020
S2	The study refined the understanding of governance and health in PSECO, analyzing classifications, strategies, and incident management. Reviewing 422 studies, it concluded that governance emphasizes innovation and competitive advantage, while incident management is still underexplored [8].	2021
S3	The study investigated and proposed governance strategies for a PSECO based on health analyses and case studies. Findings highlighted knowledge management, software quality, and innovation as key factors, with challenges such as cultural misalignment and operational constraints [9].	2021
S4	The study examined the behavior of development teams in PSECO regarding changes and incidents. Findings revealed patterns and challenges, highlighting the need for process modernization to ensure stability [11].	2023
S5	The study investigated software asset management practices in a PSECO, analyzing asset dependencies and challenges faced by IT managers. Results revealed dependency patterns and governance difficulties, emphasizing the need for risk reduction strategies to ensure platform stability [10].	2024

compared and categorized the data, extracting codes from study excerpts that were refined to be requirements for the cloud migration strategy guide;

(ii) Requirements Classification: After generating codes for each text excerpt, candidate requirements were described and classified as functional or non-functional. This classification was conducted by a team of three researchers specializing in SECO and Software Engineering (SE). Each researcher classified the requirements independently, followed by group discussions to resolve differences and reach consensus. Next, the results were reviewed by two senior researchers with over 15 years of experience in these fields. The process continued until consensus was reached on the final categorization of requirements;

(iii) Requirements Prioritization and Negotiation: Three researchers grouped initial requirements by purpose, then submitted them to two senior researchers for review. Discrepancies were resolved by consensus, resulting in a final set of 12 requirements for the guide. The complete codebook is available on Zenodo: `https://doi.org/10.5281/zenodo.15062217`;

(iv) Requirements Specification: Once the requirements were defined, we performed a structured specification process. Each requirement was documented following the Wiegers and Beatty [30] standard format, which states: *"The system must allow (accept or authorize) [user type or actor name] to [perform a specific action]"*. Tab. 2 describes an example, outlining the fields used for encoding requirements during the specification phase. The complete set of requirement specifications is available at `https://doi.org/10.5281/zenodo.15062217`;

Table 2. Coding form used in requirements specification.

Study	S2
Study Section	4.2 - Governance Maturity and Evolution
Text Fragment	*"The adoption of new architectures in proprietary SECOs must consider security, scalability, and governance to sustain the model."*
Code	Modern Architecture (R5)
Requirement Description	The guide should allow the user to indicate whether the software asset already uses or is capable of adopting a modern architecture such as microservices, serverless, or event-driven design.
Type	Functional

(v) Development of the Guide Proposal: We developed an artifact to instantiate the cloud migration strategy guide, ensuring all requirements were addressed with traceability. A decision tree was adopted as the core artifact, providing a structured and visual approach to guide IT managers in selecting strategies based on predefined criteria. It incorporates migration patterns, cost-benefit trade-offs, and governance constraints, aligning decisions with organizational goals [16];

(vi) Evaluation of the Guide Proposal: The final version of the decision tree artifact used to instantiate the cloud migration strategy guide was validated in collaboration with industry practitioners, ensuring that it meets the practical demands of a large organization operating within a PSECO.

Case Description and Diagnosis. For privacy reasons, the organization's name has been omitted and will be referred to as "X". Founded over 80 years ago, "X" is one of the largest insurance groups in Latin America, operating internationally across multiple segments, including auto, property, health, capitalization, and pension plans. The company has over 200 service centers and a network of 40,000 insurance brokers. Technical details about the company's systems were intentionally limited at the organization's request.

Over the years, "X" has accumulated a complex portfolio of legacy systems that support its core business functions, such as underwriting policies, processing claims, managing customer records, and ensuring regulatory compliance. However, as these systems age, they become architecturally degraded, reliant on obsolete technologies, and misaligned with evolving business needs, making maintenance, integration, and modernization increasingly difficult [15].

A major concern is the growing cost of maintaining these systems. Many are monolithic, tightly coupled, and depend on outdated technologies, limiting scalability and integration with modern platforms. Additionally, knowledge loss due to employee turnover further complicates maintenance efforts. To address these challenges, "X" has launched a strategic modernization program to migrate legacy systems to the cloud, aligning with industry trends that leverage cloud adoption for agility, cost reduction, and digital transformation [4]. As noticed by

Seacord et al. [22], *"software modernization attempts to evolve a legacy system when conventional maintenance and enhancement no longer suffice"*.

Despite these benefits, "X" faces hurdles in defining a cloud migration strategy due to the heterogeneity of legacy systems and decades of embedded business rules. Selecting the right approach requires careful evaluation. To support this process, we propose a systematic guide to help IT managers navigate migration complexities using research insights and industry best practices.

4 Results and Discussion

In Tab. 3 we presented the 12 requirements (identified as R1 to R12) elicited to compose the migration strategy guide based on the analysis of the five selected studies, using the research method described in Section 3. The excerpts from the studies that generated the codes for each requirement are also available at Zenodo (`https://doi.org/10.5281/zenodo.15062217`).

4.1 Main Findings

The triangulation process was carried out to ensure that the requirements derived from empirical observations align with the challenges identified in previous studies. Additionally, this process helps to strengthen the validity of the findings.

Legacy Dependencies and Architectural Constraints. Requirements R1 and R2, based on S2 and S4, highlight the challenges of outdated technologies and tightly coupled architectures. S5 reinforces it with focus on hybrid systems. R3 and R6, drawn from S2–S5, reflect SECO governance and scalability issues in proprietary assets. These studies reveal how legacy integrations hinder cloud migration. The guide offers mechanisms to assess whether dependencies (e.g., databases, APIs, mainframes) can scale within distributed architectures.

Performance and Operational Constraints. The impact of migration on system performance was a major concern identified in S5, where cloud migration strategies must consider acceptable latency levels to avoid degrading user experience. This aligns with R4, ensuring that the guide allows decision-makers to evaluate whether an asset meets predefined performance thresholds. Furthermore, R5 was derived by S2–S5, which highlighted the role of cloud-native architectures in modernization. To address this, the guide includes evaluation criteria to determine whether an asset can adopt modern paradigms such as microservices, serverless computing, or event-driven design.

Governance and Strategic Considerations. Governance aspects were addressed by S1 and S2, which examined software asset management and governance mechanisms in PSECO. The need to assess a software asset's strategic and financial value (R8) was highlighted in S1-S4, where asset evaluation emerged as a key element of IT governance in large organizations. Moreover, R7 was supported by insights from S3, which revealed that organizations struggle with defining clear decommissioning strategies for legacy systems. The guide ensures

Table 3. Requirements and decision criterion elicited for the guide.

ID	Requirement	Description	Type	Study	Decision Criteria (DC)	ID DC
R1	Deprecated version	The guide should allow the user to indicate whether the software asset depends on any other asset or package that is in a deprecated version.	Functional	S2, S4	Does the solution depend on any software or package that is in a deprecated version?	DC1
R2	Coupling	The guide should allow the user to indicate whether the software asset has an on-premises architecture that is coupled with existing applications.	Functional	S1, S4, S5	Does the solution have coupled (on-premises) architecture?	DC2
R3	Application integration	The guide should allow the user to indicate whether the software asset require integrations or depends on other assets in the on-premises infrastructure.	Functional	S2, S3, S4, S5	Does the solution require integrations and be dependent on others that are in the on-premises infrastructure?	DC3
R4	Latency Level	The guide should allow the user to indicate whether the software asset meets the acceptable latency level for integrations, considering performance requirements and the impact on user experience.	Functional	S5	Does the system meet the acceptable latency level for integration and performance requirements?	DC4
R5	Modern Architecture	The guide should allow the user to indicate whether the software asset already uses or is capable of adopting a modern architecture based on microservices, serverless, or event-driven design.	Functional	S2, S3, S4, S5	Is the solution already using or capable of adopting modern architectural paradigms (e.g., microservices)?	DC5
R6	Scalability	The guide should allow the user to indicate whether the software asset's dependencies, such as databases, mainframes, and external APIs, are scalable and compatible with distributed environments.	Functional	S2, S3, S4	Do the solution and its dependencies have high complexity for scalability (e.g., mainframe)?	DC6
R7	Discontinuation	The guide should allow the user to indicate whether there is a formal discontinuation plan for the software asset before the migration deadline and what actions should be taken.	Functional	S3	Is there a formal discontinuation plan for the software asset before the migration deadline?	DC7
R8	Strategic and Financial Value	The guide should allow the user to indicate whether the software asset generates revenue or holds strategic value for the company.	Functional	S1, S2, S3, S4	Does the application generate revenue or hold strategic value for the company?	DC8
R9	On-Premises Transformation	The guide should allow the user to specify whether the software asset is undergoing a significant modernization process within the on-premises environment.	Functional	S3, S4	Is the application undergoing significant on-premises modernization?	DC9
R10	Hybrid Approach Support	The guide should allow the user to indicate whether the software asset supports a hybrid approach, meaning it enables part of the solution to remain on-premises while the other part is in the cloud.	Functional	S1, S2	Does the application support a hybrid approach, maintaining part of the system on-premises and part in the cloud?	DC10
R11	Contractual restriction	The guide should allow the user to indicate whether the software asset has contractual restrictions or licensing requirements that prevent the solution from being run in a public cloud under a pay-as-you-go model.	Functional	S2, S3	Does the software have licensing or contractual restrictions that prevent its migration to a public cloud model?	DC11
R12	Platform availability	The guide should allow the user to indicate whether the technologies, products, and packages required for the software asset are available as SaaS[a] on a cloud platform.	Functional	S1, S2, S3, S4, S5	Are the required packages available as a SaaS solution on a cloud platform?	DC12

[a] Software as a Service (SaaS) is a cloud computing model that enables the use of online applications without the need for local installation, such as Google Drive and Dropbox.

that migration decisions consider whether an asset has a formal plan for retirement or replacement.

Hybrid Migration Approaches and Regulatory Constraints. R10, from S1 and S2, reflects the need for hybrid strategies, combining on-premises and cloud components. R11, based on S2 and S3, highlights licensing restrictions in pay-as-you-go models, stressing compliance in migration planning. R12, supported by S1–S5, reinforces the need to verify cloud provider support for required technologies before migration.

4.2 Implications for Practitioners and Researchers

A key outcome of this work is the development of an artifact that instantiates the cloud migration strategy guide, ensuring systematic incorporation of all defined requirements. As shown in Fig. 2, this decision tree provides a structured approach to evaluate software assets and select an appropriate migration strategy. In Tab. 3 we have further illustrated how the requirements align with the decision criteria (DC), operationalizing the migration assessment process. Each decision tree node represents a DC from previous studies. By answering questions on constraints, IT managers can identify the best cloud migration strategy. The work also links migration strategies to software asset management in PSECO.

The decision tree was developed based on real challenges from a participative case study in a large organization with complex legacy systems. Issues such as tight mainframe coupling, architectural obsolescence, and knowledge loss shaped the decision criteria. The artifact combines theoretical insights with practical constraints, linking each criterion to governance, legacy limitations, and sustainability goals. This makes it useful for structured, justifiable cloud migration decisions and offers a replicable framework for researchers.

4.3 Operationalizing the Migration Decision Process

The decision tree simplifies complex migration decisions by structuring them into a sequence of well-defined evaluation points. For example, in our context, the organization "X" operates a J2EE-based application that manages critical insurance policies. This system relies on EJB (Enterprise JavaBeans) components to encapsulate business logic and integrates with mainframe programs running on IBM CICS (Customer Information Control System) and IBM Db2 (Database 2nd Gen). The decision tree guides the evaluation process as follows:

Is the solution already using or capable of adopting modern architectural paradigms (e.g., microservices)? → No (DC5); Is there a formal discontinuation plan for the software asset before the migration deadline? → No (DC7); Does the application generate revenue or hold strategic value for the company? → Yes (DC8); Does the solution depend on any software or package that is in a deprecated version? → No (DC1); and Does the application and its dependencies have high complexity for scalability (e.g., mainframe)? → Yes (DC6).

As the system uses supported versions, has manageable scalability, and lacks modern architecture, the recommended strategy is *Refactor*, enabling gradual

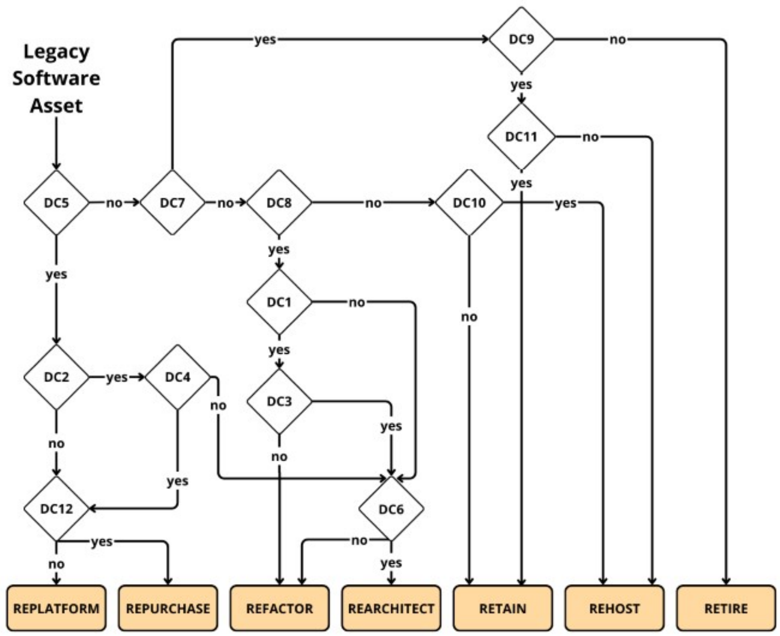

Fig. 2. Cloud migration strategy guide through a decision tree.

modernization with minimal disruption [23]. If dependencies were more complex, a *Rearchitect* approach would be more appropriate [5].

Finally, we engaged two IT managers and five senior analysts to evaluate the decision tree. They applied it to ten legacy assets from the PSECO platform. Feedback was gathered via a five-point Likert-scale evaluation question (EQ) and an open question (OQ) on confidence in migration decisions. Results showed a positive overall perception, helping validate and refine the guide.

For **EQ**, all practitioners (100%, 7 of 7) strongly agreed, indicating that the decision tree effectively supports IT managers in selecting cloud migration strategies for legacy software assets. This structured approach not only enhances the consistency of decisions, but also provides transparency in migration choices, reducing uncertainty, and facilitating discussions among stakeholders, software architects, and IT management teams.

Practitioners also provided feedback on the **OQ**, highlighting the potential of the decision tree artifact to guide decision-making processes toward the most appropriate migration strategy. However, P7 (IT manager) stated: *"Although the decision tree suggested 'replatform', I chose 'rearchitect' to leverage my team's expertise as early adopters and enhance our visibility at the executive level. This decision reinforces our role in driving innovation within the company"*. The practitioners' feedback is available at `https://doi.org/10.5281/zenodo.15062217`

Therefore, the cloud migration decision was influenced by considerations beyond technical factors, incorporating external business drivers. Therefore, our findings revealed that a one-size-fits-all approach to cloud migration is impractical, as each legacy software asset has distinct dependencies, performance requirements, and regulatory constraints.

5 Limitations and Threats to Validity

Although the elicitation of requirements was based on findings from five selected studies covering both theoretical and practical perspectives, one limitation was the initial set of requirements reflected a specific context of the PSECO. Future investigations considering different organizational contexts and cloud adoption maturity levels may refine the set of requirements for broader applicability.

To reduce risks to internal validity, the requirement elicitation process followed a systematic and reproducible approach. Additionally, the decision tree artifact was iteratively validated by practitioners, whose expertise helped refine the migration criteria and align them with real-world decision-making processes. Regarding external validity, which concerns the generalizability of our findings, we sought to evaluate our guide in a real-world enterprise environment. This ensures that the proposed migration strategies reflect practical constraints and decision-making dynamics observed in large-scale software asset modernization.

6 Conclusion

This work explored cloud migration strategies for legacy software assets in a PSECO and proposed a structured decision-making approach through a decision tree artifact. By conducting a participative case study in a large international organization, we identified key challenges IT managers face when modernizing legacy systems while maintaining governance and operational stability. The results indicated that migration decisions should balance multiple factors, including technical sustainability, business strategies, and compliance requirements.

The decision tree artifact developed in our work provides a structured approach to guide IT teams in selecting the most appropriate migration strategy based on the organization's governance model, software dependencies, performance, constraints, and long-term business objectives. The findings revealed that organizations with well-defined governance policies and structured migration strategies can minimize technical debt and improve the resilience of the technological platform. Future work may focus on refining governance models for cloud migration in PSECO, exploring automated decision-support tools, and analyzing the impact of migration strategies on business competitiveness.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001 and Grants 88887.959659/2024-00 and 88887.955223/2024-00; CNPq (Grant 316510/2023-8); FAPERJ (Grant E-26/204.404/2024); and UNIRIO.

References

1. Ahmad, N., Naveed, Q.N., Hoda, N.: Strategy and procedures for migration to the cloud computing. In: 2018 IEEE 5th international conference on engineering technologies and applied sciences (ICETAS). pp. 1–5. IEEE (2018)
2. Alves, C., Oliveira, J., Jansen, S.: Understanding governance mechanisms and health in software ecosystems: A systematic literature review. In: International Conference on Enterprise Information Systems. pp. 517–542. Springer (2017)
3. Andrikopoulos, V., Lago, P.: Software sustainability in the age of everything as a service. Next-Gen Digital Services. A Retrospective and Roadmap for Service Computing of the Future: Essays Dedicated to Michael Papazoglou on the Occasion of His 65th Birthday and His Retirement pp. 35–47 (2021)
4. Assunção, W.K., Marchezan, L., Arkoh, L., Egyed, A., Ramler, R.: Contemporary software modernization: Strategies, driving forces, and research opportunities. ACM Transactions on Software Engineering and Methodology (2024)
5. Chinamanagonda, S.: Cloud migration strategies and best practices. Available at SSRN 4986770 (2019)
6. Corbin, J., Strauss, A.: Basics of qualitative research: Techniques and procedures for developing grounded theory. Sage publications (2014)
7. Costa, L.A., Fontão, A., Santos, R.: Investigating asset governance mechanisms in a proprietary software ecosystem. In: Proceedings of the XVI Brazilian Symposium on Information Systems. pp. 1–8 (2020)
8. Costa, L.A., Fontão, A., Santos, R.: Investigating proprietary software ecosystem governance and health: An updated and refined perspective. In: XVII Brazilian Symposium on Information Systems. pp. 1–8 (2021)
9. Costa, L.A., Fontão, A., Santos, R.: Toward proprietary software ecosystem governance strategies based on health metrics. IEEE Transactions on Engineering Management **69**(6), 3589–3603 (2021)
10. Costa, L.A., Outão, J.C., Simões, J.E., Santos, R.P.d.: Towards software asset management in proprietary software ecosystems: A participative case study. In: Proceedings of the XXIII Brazilian Symposium on Software Quality. pp. 69–80 (2024)
11. Costa, L.A., Sampaio, A., Fontao, A., dos Santos, R.P.: Investigating change and incident management in a proprietary software ecosystem: Survival mode on. In: Anais do XII Brazilian Workshop on Social Network Analysis and Mining. pp. 55–66. SBC (2023)
12. Dhungana, D., Groher, I., Schludermann, E., Biffl, S.: Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume. pp. 96–102 (2010)
13. Hilty, L.M., Arnfalk, P., Erdmann, L., Goodman, J., Lehmann, M., Wäger, P.A.: The relevance of information and communication technologies for environmental sustainability–a prospective simulation study. Environmental Modelling & Software **21**(11), 1618–1629 (2006)
14. Huseynli, M., Papp, A., Bub, U., Ogbuachi, M.C.: Interaction room for cloud migration (ir: Cm): A systematic approach. In: ICEIS (2). pp. 203–210 (2023)
15. Kiran Mallidi, R., Sharma, M., Singh, J.: Legacy digital transformation: Tco and roi analysis. International journal of electrical and computer engineering systems **12**(3), 163–170 (2021)

16. Korting, T.S.: C4. 5 algorithm and multivariate decision trees. Image processing division, national institute for space research–INPE Sao Jose dos Campos–SP, Brazil **22** (2006)
17. Lago, P., Koçak, S.A., Crnkovic, I., Penzenstadler, B.: Framing sustainability as a property of software quality. Communications of the ACM **58**(10), 70–78 (2015)
18. Lenarduzzi, V., Lomio, F., Saarimäki, N., Taibi, D.: Does migrating a monolithic system to microservices decrease the technical debt? Journal of Systems and Software **169**, 110710 (2020)
19. Manikas, K.: Revisiting software ecosystems research: A longitudinal literature study. Journal of Systems and Software **117**, 84–103 (2016)
20. Manikas, K., Hansen, K.M.: Software ecosystems–a systematic literature review. Journal of Systems and Software **86**(5), 1294–1306 (2013)
21. Santos, R.P.: Managing and monitoring software ecosystem to support demand and solution analysis. Ph.D. thesis, Universidade Federal do Rio de Janeiro (2016)
22. Seacord, R.C., Plakosh, D., Lewis, G.A.: Modernizing legacy systems: software technologies, engineering processes, and business practices. Addison-Wesley Professional (2003)
23. Shastry, A.L., Nair, D.S., Prathima, B., Ramya, C.P., Hallymysore, P.: Approaches for migrating non cloud-native applications to the cloud. In: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0632–0638. IEEE (2022)
24. Shuaib, M., Samad, A., Alam, S., Siddiqui, S.T.: Why adopting cloud is still a challenge? a review on issues and challenges for cloud migration in organizations. Ambient Communications and Computer Systems: RACCCS-2018 pp. 387–399 (2019)
25. Sommerville, I.: Software Engineering, 9/E. Pearson Education India (2011)
26. Teece, D.J.: Dynamic capabilities and strategic management: Organizing for innovation and growth. Oxford University Press, USA (2009)
27. Trinkenreich, B., Santos, G., Barcellos, M.P., Conte, T.: Combining gqm+ strategies and okr-preliminary results from a participative case study in industry. In: International Conference on Product-Focused Software Process Improvement. pp. 103–111. Springer (2019)
28. Vaka, D.K.: The sap s/4hana migration roadmap: From planning to execution. Journal of Scientific and Engineering Research **11**(6), 46–54 (2024)
29. Verdecchia, R., Lago, P., De Vries, C.: The future of sustainable digital infrastructures: A landscape of solutions, adoption factors, impediments, open problems, and scenarios. Sustainable Computing: Informatics and Systems **35**, 100767 (2022)
30. Wiegers, K.E., Beatty, J.: Software requirements. Pearson Education (2013)
31. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering. Springer Science & Business Media (2012)
32. Zacarias, R., Parracho, T., Moura, F., Lima, D., Santos, S., Rodrigues, K., Seruffo, M., Santos, R.: Uma proposta de framework multimodal para avaliaçao de experiência de usuário em sistemas web interativos. In: Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software (WASHES). pp. 94–105. SBC (2024)
33. Zacarias, R.O., Santos, R.P.d., Lago, P.: Exploring transparency as a sustainability goal in software ecosystems. In: Proceedings of the 12th ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems. pp. 45–52 (2024)