Navigating the Modernization of Legacy Applications and Data: Effective Strategies and Best Practices

ABSTRACT

Aims: This study aims to provide a deep-dive analysis of organizations' challenges when modernizing legacy applications. In a broader concept, contemporary research identifies the key drivers for modernization, enumerates the challenges, and offers actionable strategies and best practices for overcoming them.
Study design: The study employs a multi-method approach that combines a literature review, examples of case studies, and interviews with IT professionals experienced in legacy application modernization.
Place and Duration of Study: This global study was conducted over three years, from January 2020 to August 2023.
Methodology: This research uses a literature review to collect data. In the literature review process, a comprehensive array of data collection methods is strategically employed to ensure the acquisition of a diverse and pertinent body of knowledge concerning the challenges associated with modernizing legacy applications and the effective strategies and best practices to address them. It starts with searching in extensive Online Databases and Repositories, using Keyword Searches and Citation Tracking to find the relevant literature. Systematic reviews and meta-analyses give structured synthesis, while manual searches collect real-world case studies. Grey Literature supplements insights, and Evidence-Based Practices (EBP) ensure rigor. Thematic Analysis sorts findings, whereas Data Management arranges data, and The CAP evaluates the credibility of sources. This approach is an important starting point for modernizing legacy systems and developing effective policies and guidelines.
Results: This study cites two primary drivers of modernization: business needs and technology. It also outlines challenges organizations face during modernization, like data migration, complex legacy stacks, user acceptance and integration. Furthermore, the study looks at various strategies to overcome these challenges, like the appropriate modernization approach and best practices.
Conclusion: The study concludes that, while modernizing legacy applications is fraught with challenges, these can be overcome with the right planning, strategy, and execution. Organizations can transform their legacy systems into assets more aligned with modern business needs and technological capabilities.

Keywords: Legacy application modernization, Data Modernization, Modernization Strategies, Data migration and User acceptance.

1. INTRODUCTION

The modernization of legacy computational systems in computer science is an area which has been changing for a long time. These systems typically support essential research or practical solutions components, which, however, require more and more effort to maintain, extend, and evolve with changing needs. A legacy system is any computing software, hardware, or combination derived from previous language, platform, and methodology generations. While such systems usually keep functioning and fulfil some purposes within the enterprise, they tend to degrade over time in terms of their reliability, maintainability, and interoperability with new technology trends. There are multiple reasons why it is necessary to modernize: As old systems age, they become more at risk due to modern cyber-attacks, which were never intended to be prevented by the system's initial architecture. They are typically unable to scale with growing business needs, resulting in operational bottlenecks and impeding growth. As vendors stop supporting the old systems, the enterprises become unsupported for updates, making them increasingly obsolete to the newest tech.

On the other hand, legacy systems tend to weigh organizations down. They generally necessitate expertise to run, which costs more and can become rare when the tech becomes obsolete. In addition, these systems may not be easily integrated with other new technologies resulting in data silos and inoperative inefficiencies. Thus, modernization cannot be considered an option but a need. It leverages emerging innovations, like cloud computing, analytics and AI, improving operating costs, scalability, and organizational flexibility. However, that is not the point; the objective is for them to move from liability to an asset in driving business value and innovation.

Legacy systems are often the backbone of an organization's IT infrastructure, serving critical functions that keep operations running smoothly. However, these systems, often decades old, face challenges such as a lack of compatibility with newer technologies, scalability issues, security vulnerabilities, and high maintenance costs. As technology evolves rapidly, organizations find themselves at a crossroads: to modernize these legacy systems or continue to pour resources into increasingly obsolete technology.

The crux of the issue lies in the need for legacy systems to adapt to modern-day requirements and technologies. These outdated systems are often inflexible, not scalable, and need more essential features for implementing modern business processes. This leads to reduced competitiveness in today's digital marketplace, increased risk of security breaches, and rising costs due to maintenance and compliance challenges.

The proposed solution involves adopting a multi-faceted approach to modernize legacy systems, with strategies tailored to each organization's unique circumstances. These strategies could range from encapsulation, where the legacy system is made to interact with newer technologies, to complete re-architecture, where the system is overhauled to align with current needs and future scalability. Alongside these strategies, implementing best practices like comprehensive assessments, clear requirements, effective change management, and rigorous testing can contribute to a smoother and more effective modernization process.

Numerous studies have explored legacy system modernization. Research in this area often delves into methodologies like "rehosting" or "refactoring" as viable solutions [1]. Meanwhile, studies like Johnson and Miller [2] focus on the ROI aspect of modernization, concluding that the long-term benefits often outweigh the short-term costs. Recent literature has also highlighted the importance of stakeholder buy-in and effective change management [3]. However, there is a gap in comprehensive guides that cover not just the strategies but also best practices and the human aspect of the modernization process, including the change management required to make the transition successful.

This study applies to organizations spanning multiple industries that depend upon outdated systems for their routine activities. The justification for this work is two-fold:
1. Practical Relevance: In today's digital world, many organizations still face the challenge of modernizing legacy systems. Therefore, this work will be the ultimate guide (blueprint) for such companies, organizations, and industries.
2. Academic Significance: This work is indispensable to academia because it is an illumination that unifies and organizes scattered knowledge and information on challenges in the modernization of legacy applications - strategies and best practices, thus filling in the gaps in the existing literature. This can be important for researchers, scholars or anyone who wants to understand this topic better.
This work offers a complete package that organizations can use as a reference guide by addressing the challenges, strategies, and best practices for legacy system modernization. A multi-faceted approach provides an actionable plan for tackling this critical issue, thus enabling organizations to stay competitive in today's fast-paced digital world.

2. METHODOLOGY

The imperative for legacy application modernization is increasingly becoming a strategic necessity catalyzed by business dynamics and technological evolution. As the business landscape becomes more competitive, organizations are under immense pressure to adapt quickly, innovate, and meet customer expectations. Legacy systems, however, are often bottlenecks to this agility. They need scalability to cope with increasing volumes of data and transactions, and their inflexible architectures make it difficult to integrate new functionalities or adapt to changing business models. Simultaneously, technological advancements are reshaping the possibilities for enterprise operations. The rise of cloud computing, machine learning, data analytics, and cybersecurity measures offer tools and capabilities incompatible with dated, monolithic legacy systems. Failure to modernize can result in increased operational costs, security vulnerabilities, and missed opportunities for leveraging technology to gain a competitive edge. In summary, the need for legacy application modernization is driven by an urgent requirement to align IT capabilities with evolving business strategies and to exploit emerging technologies for operational excellence. This makes modernization not just an IT concern but a critical business imperative.

2.1 BUSINES DRIVERS

The global business landscape is incredibly dynamic, with new competitors, new customer expectations, and new global challenges emerging rapidly. In such an environment, agility, scalability and innovation are the survival and growth. Legacy systems, unfortunately, act as a bottleneck in these areas [1]. For instance, older systems often need more features and flexibility to integrate with modern e-commerce platforms and customer relationship management (C.R.M.) systems, essential to a high-quality customer experience. This may also mean that the organization's ability to pivot or scale effectively may need to improve as the organization tries to move towards new business models and processes[2].

Moreover, maintaining legacy systems usually costs higher operational costs. These systems often require specialized maintenance skills that are increasingly hard to find and expensive. There are the downtime costs, which in today's 24/7 business world can be particularly damaging. Lastly, legacy systems can impact decision-making. They generally need more advanced analytics and real-time data processing capabilities for quick decisions. The result is often missed opportunities and reduced competitiveness [3]. For these reasons, the business reasons behind legacy application modernization are compelling. By modernizing,

organizations can better align their I.T. strategy with their goals, reduce costs, and increase their agility and responsiveness to market changes [4].

2.2 TECHNOLOGY/IT DRIVERS

With the advent of technologies like cloud computing, the Internet of Things (IoT) and artificial intelligence (AI), not business alone, but everything is being transformed. The cloud, IoT, artificial intelligence (AI), and block chain are rapidly changing the way business used to be run. Legacy systems must be equipped to effectively leverage these technologies, leading to a gap between what the technology can offer and what the business can utilize [5]

a) Older systems were not designed to withstand today's cybersecurity challenges, making them more vulnerable. This vulnerability poses a significant risk, considering cyber threats' increasing frequency and sophistication [6].

b) As new software solutions emerge, the need for more support for older systems becomes more apparent. Vendors often discontinue support for older software versions, making resolving issues or integrating with newer systems difficult [7].

c) The inefficiency of legacy systems can drain the organization.

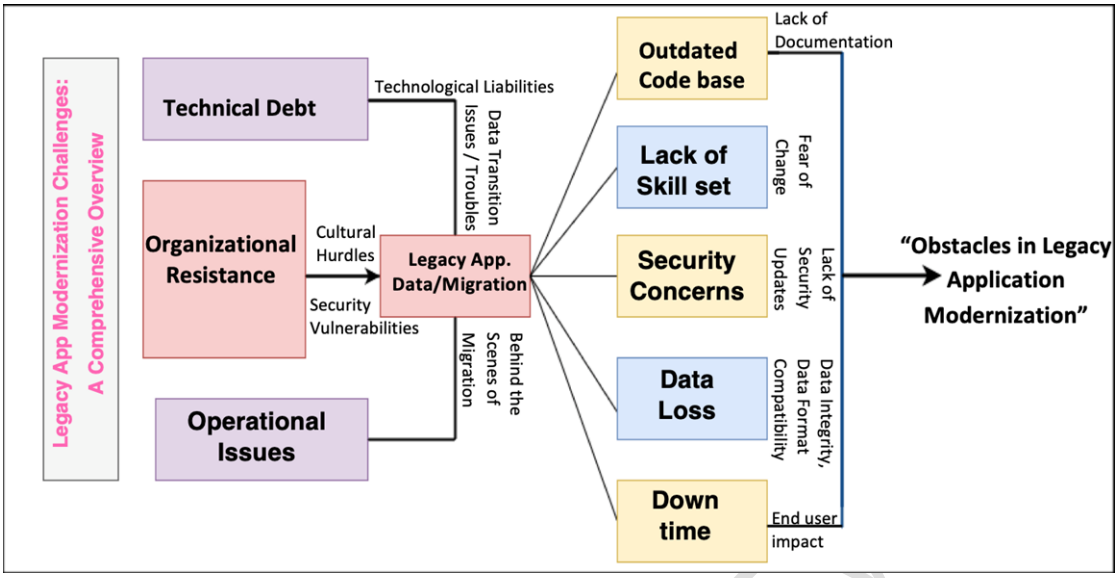
They often need to be more compatible with modern, more efficient technologies like virtualization and containerization, which can optimize resource usage and reduce costs [8]. Technological obsolescence also affects an organization's ability to attract and retain talent. The new generation of IT professionals is trained in modern technologies and may need help to work with outdated systems [9].

Therefore, from a technological standpoint, the modernization of legacy applications is not just about keeping up with the latest trends; it's about future-proofing the organization. It ensures that the business can fully leverage new technologies for improved efficiency, security, and innovation [10].

2.3 LEGACY SYSTEM MODERNIZATION CHALLENGES

Modernizing legacy systems is a complex endeavor fraught with numerous challenges that can significantly impact the project's success. While the end goal is to achieve a more agile, efficient, and secure infrastructure, organizations often grapple with technical intricacies and operational issues. These challenges range from data migration versus application migration dilemmas to user acceptance, integration hurdles, and strategic decision-making. This section delves into one of the most critical aspects— the challenges concerning data migration’sV.s application's migration's [11].

Figure 1: Challenges in Modernization of Legacy Applications



2.3.1 Data-Migration V.s Application-Migration

Data migration and Application migration are separate but equally important elements of the modernization journey, with their hurdles [12]

2.3.2 Data Migration

Data Migration is when data is migrated from an old legacy database/system to a new one. All those things come to mind when talking about this process, which is complex, including data format variations, data quality errors, and data loss/corruption risks. Time is money when processing such a vast amount of data. Additionally, moving data without confirming its veracity might result in non-compliance issues (especially in case the data is sensitive or regulated) [13].

2.3.3 Application Migration

Here, application migration targets applications where data is processed and managed. This type of migration can be complicated due to outdated code, dependencies on obsolete technologies, and potential compatibility issues with other systems in the organization. There's also the challenge of ensuring that the newly migrated application interfaces seamlessly with existing systems to prevent data silos or operational inefficiencies [14].

2.3.4Balancing the Two

The real challenge lies in proper balance. Both are interconnected. A failure in one can lead to the other. For example, a poorly executed data migration can render a successfully migrated application useless and vice versa. Hence, a well-thought migration strategy that considers data and application aspects, migration and transition, and operational continuity is crucial for a migration strategy [15]

2.4 LEGACY DATA STACKS ARE COMPLEX, INEFFICIENT, AND COSTLY

Legacy data stacks are often the primary challenge in modernization due to their inherent complexities, inefficiencies, and costs. These data stacks tend to be based on ageing database management systems and storage architectures that need to be equipped for the demands of today's high velocity, high volume data [16].

2.4.1 Complexity

A legacy data stack can often be complicated, with multiple layers and components bolted on through the years. Consequently, the result is an opaque, complex system to be handled. Data complexity is not only a hurdle for migration but also increases the likelihood of errors and corruption [17].

2.4.2 Inefficiency

Legacy data stacks are infamous for their inefficiency. Moreover, they cannot handle indexing, real-time analytics, or fast data retrieval, things we take for granted with modern database management systems. Inefficiencies in the system can hinder business processes dramatically and may prevent an organization from losing its competitive advantage [18].

2.4.3Cost

Keeping a legacy data stack can be an expensive, soon costly affair. They usually require specialized skills to manage and operate, and as these systems age, finding experts who understand them becomes increasingly difficult and expensive. Furthermore, these older systems often consume more power and occupy more physical space than their modern counterparts, adding to the overall operational costs [19].

To sum up, legacy data stacks are a complex web of technologies that are costly to maintain. Modernization is not just viewed as merely a technical upgrade but a business imperative to streamline operations, improve decision-making, and reduce costs [20].

2.5 USER ACCEPTANCE

User acceptance is one of the often overlooked yet critical challenges in legacy modernization. Employees using a legacy system for years may resist change, primarily because of the learning curve associated with new systems. This resistance can lower productivity, morale, and project failure if not managed adequately [21].

2.5.1 Psychological Barriers

People are generally averse to change, especially when it involves a tool or system they use daily. This psychological barrier can be a significant barrier to modernization. Employees might be concerned about how the new system will impact their work routines and how they will adapt to it [22].

2.5.2 Training and Onboarding

Effective training programs can alleviate some of these concerns. However, training is more than just teaching employees how to use the system. It is also about helping them understand its benefits to their work and organization. Proper onboarding processes, including comprehensive training modules and readily available support, can go a long way in easing the transition [23].

2.5.3 Change Management

Strategic change management is crucial for successful user acceptance. Regular communication about why the change is happening, the benefits, and how it will affect individual roles can help set the right expectations. Feedback loops should also be established to address concerns and issues promptly [24].

In summary, user acceptance is a multifaceted challenge beyond mere training. It involves breaking down psychological barriers, providing robust training, and effective change management to ensure that the modernization effort is both a technical success and a human one [25].

2.6 SEAMLESS INTEGRATION OF NEW APPLICATIONS AND TECHNOLOGIES

Integrating new applications and technologies is crucial in modernizing a legacy system. Integration may create operational inefficiencies, siloed data, and potential financial risks if data is lost during integration [26].

2.6.1 Technical Challenges

The integration piece is generally more complicated because it typically means that we must ensure that everything integrates well with the old (legacy) systems being replaced. New applications may demand new data formats or communication protocols involving data transformation and adaptation operations that take up much of the developer's time and are prone to errors [27].

2.6.2 Data Loss and its Costs

The risk is that one could lose information and suffer a fortune at the integration stage. Data loss can disrupt business continuity, diminish clients' credibility and infringe upon lawful boundaries, especially where data is deemed personally confidential or regulated; data lost information cost, whether temporary or permanent, can impact business reputation and expansion.

2.6.3 Mitigating Risks

A properly designed integration plan helps overcome the hurdles. Such a strategy would involve thorough testing to determine the potential for data loss or incompatibility problems before full-scale implementation. As well as disaster recovery measures must be established to recover the data if any data is lost.

2.6.4 Role of DevOps and CI/CD

DevOps can also emerge from modernized DevOps practices and CI/CD pipelines that support them. These approaches facilitate small-step improvements, constant testing, and smooth rollbacks when things go wrong; this makes the whole deployment process easier to deal with and more robust.

Lastly, user can only do with the integration piece for modernization projects. This can only be achieved with diligent planning, rigorous testing and adherence to advanced software development practices to reduce any chance of losses (costs) from data loss [30].

2.7 DIFFICULTIES WITH CHOOSING THE RIGHT STRATEGY

Choosing an optimal modernization strategy for legacy applications is a critical move that intertwines with numerous complexities and far-reaching consequences in the direction of the entire project. This process has become more complex when aligning strategy with organizational goals while carefully allocating budgets considering timeframes and technical limitations [31]. One right move, such as an accurate strategy selection, can translate into numerous negative impacts like delayed projects and high costs, ultimately leading to project failure. This highlights why it is critical to choose an appropriate strategy that would keep the project safe from possible fault-finding and guide it towards its intended goals [31].

2.7.1 Multiple Options, Multiple Challenges

Modernization landscapes provide organizations with numerous strategies, including comprehensive system makeovers such as re-architecting and agile approaches to modernization. The decision-making process for these strategies is complex because each has pros and cons. For example, a company may revamp all the systems, which can be expensive and time-consuming, but it is worth it in the future. An agile modernization methodology can bring immediate wins but may only provide some of the benefits of a comprehensive modernization. The fact is that organizations have to be very much aware of their objectives, limits and trade-offs between short-term profits and potential long-term benefits when determining a modernization strategy [32].

2.7.2 Stakeholder Alignment

Another issue is aligning different project stakeholders with their interests. Business leadership may focus more on short-term success and cost reduction, while the IT team focuses on long-term strength and flexibility. A solution that pleases everyone is often not simplistic.

2.7.3 Risk Assessment

Accurate diagnosis of the problem is hard but critical. As with any decision, there is always a trade-off — balancing potential benefits versus risks for each approach. Organizations must consider the pros and cons of each method. This list may include market, credit, operational, and project risks for technological complexity or an absence of internal competencies [10].

2.7.4 Information Overload

The flood of data and pundits may also overwhelm humans, causing them to get analysis paralysis. Organizations may need assistance in choosing the right choice [11].

The best transformation approach is a complex decision based on knowledge of the organization's needs and technical potential [13]. This typically entails the following steps: strategic planning; engagement of key stakeholders; identification, quantification, and mitigation of potential risks; development of relevant information for decision-making; identification and evaluation of alternatives [7].

2.8 CHALLENGES WITH DATA MIGRATIONS

Data migration is an essential phase during the legacy system modernization process, with specific problems that affect the whole transformation's quality greatly. This challenge can be broken down into several major categories.

2.8.1 Poor Knowledge of Source Data/Source Code

Organizations often need a complete picture of their legacy systems' source data and code. As such, organizations might have lost information over time due to employee attrition or retirement. The information must be included to ensure accurate data mapping, leading to data consistency. Enterprises must truly grasp the underlying system's source data and source code in such situations. Any ignorance can create data mapping issues that result in inconsistent data and errors during the migration [12].

2.8.2 Unclear Requirements

Confusing or ever-changing requirements can become a big obstacle for Data Migration. Suppose the project team needs an accurate understanding of what the new system is

designed to achieve. It will likely cause misalignment between legacy data and the new system, leading to inefficiencies and poor data quality [19].

2.8.3 Failure to Validate the Implementation

Paying attention to the validation of a new system's implementation is critical. Following migration, rigorous validation is necessary to validate end-to-end functionality; otherwise, the system may lead to data loss or data quality problems. The validation exercise embraces several activities, such as thorough testing to confirm the data integrity, consistency, and conformity with the set business rules. For instance, information loss, data corruption, and problems with data quality can happen if comprehensive data verification is missing [15]. The validation process should include data integrity, completeness, and confirmation that the migrated data meets the business objectives [18]. Validate in this manner to avoid operational disruptions and jeopardize the success of migration efforts.

2.8.4 Incomplete or Inconsistent Data

Legacy systems often need more complete or consistent data due to years of use and maintenance. This data may not conform to modern data standards, making it difficult to migrate seamlessly. The need for data cleansing and transformation adds complexity to the migration process [21].

2.8.5 Lack of Documentation

Legacy system documentation usually needs to be included. Poorly documented data structures, dependency graphs and business rules open the possibility of errors when relying on reverse engineering and guessing [25].

These challenges require careful planning, detailed Documentation, clear requirements, and extensive testing. Organizations should also consider looking outside for expertise to safely and securely navigate the intricacies of data migration. Effective data migration is critical to guarantee that the modernized system can run accurately with clean, trustworthy and consistently formatted data – vital factors enabling business continuity and expansion [27].

3. HOW TO OVERCOME CHALLENGES: STRATEGIES AND BEST PRACTICES

To tackle these challenges effectively, legacy application modernization must be addressed through a hybrid approach using strategic methods and industry best practices. These steps are essential for advancing more advanced methods [2].

3.1 LEGACY APPLICATION MODERNIZATION OPTIONS

The strategy for modernization and development has an incredible amount of influence on the success of a project. Companies must determine what is best for their particular circumstances. Some common modernization options include [10]:

3.1.1 Encapsulation

This can include encapsulating all the components of the old system into a single package, exposing them as SOAP services. This strategy enables companies to use the legacy system's capability and develop new functionality over time [12].

3.1.2 Rehosting

Rehosting *(often "lift and shift")* is migrating legacy application onto a cloud-native infrastructure with minimal modifications to the code base. This is an easy way to start, but it might take only some modernization-enabled capabilities.

3.1.3 Replatforming

Replatforming means making small adjustments to your old application to work in the latest infrastructure, for example, moving out of the data center into the cloud. This methodology provides enhanced scaling and flexibility [14].

3.1.4 Refactoring

Refactoring the application means rewriting it from scratch — often moving from a monolith to a microservices architecture, using new design patterns. This strategy provides maximum future benefit but demands substantial upfront costs [16].

3.1.5 Re-architecture

Re-architecting the application involves a complete overhaul, often transitioning from monolithic to microservices architecture or adopting modern design patterns. This approach offers long-term benefits but requires significant investment [16].

In practice, this mix of tactics is employed all the time by businesses. For instance, encapsulating the core components of legacy systems during Rehost/Replatform and migrating the least important aspects to the cloud provides a way to optimize benefits while limiting exposure to risks [17].

3.2 OVERCOME CHALLENGES: STRATEGIES AND BEST PRACTICES

Organizations need to adopt a systematic approach along with best practices for legacy application modernization. There, we outline top approaches and best practices which could enable an easier and more effective transformation experience [18].

3.2.1 Strategies

Comprehensive Assessment: Start with an audit of the existing legacy system. It understands an application's architecture, dependencies, data, and business process flow. This assessment is the basis for making decisions on modernization strategy [19].

3.2.1.1 Clear Requirements

State the requirements/expectations for the modernization initiative. Make sure all key stakeholders are aware of the expected benefits that the new system will offer. The benefit of having detailed requirements is that it prevents misinterpretations and scope creep [20].

3.2.1.2 Effective Change Management

Implement agile change management tactics to overcome potential end-user resistance and guarantee a smooth go-live. Engage with end users early and often, educate through formal and informal training, and offer ongoing support throughout and post-modernization [21].

3.1.2.3 Rigorous Testing and Validation

Testing is the most important part of modernization. Rigorously tests each module (including data migration, feature, and integration). This validates whether the modernized system meets SLO, Security, and compliance standards [22].

3.2.2 Best Practices

3.2.2.1 Documentation

Maintain up-to-date Documentation throughout the modernization project. Documentation aids in understanding the system's architecture, configurations, and processes. It is also a valuable resource for future maintenance and troubleshooting [25].

3.2.2.2 Incremental Approach

Think about embracing an incremental modernization strategy. Incrementally transform — users do not have to move in one large leap. This enables faster wins and reduces disruption during day-to-day operations. It could mean updating individual modules/functionalities while the legacy system still exists.

3.2.2.3 Monitoring and Maintenance

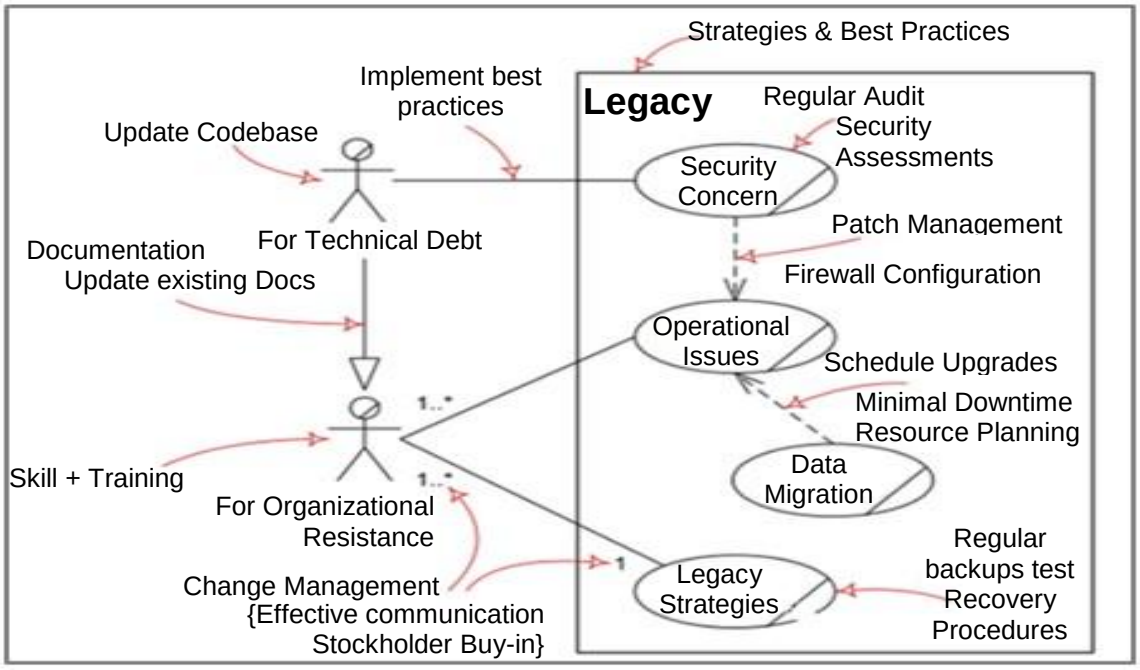
Implement post-implementation monitoring and maintenance practices. Continuously monitor the modernized system's performance, security, and user satisfaction. Be prepared to address any issues promptly to ensure the system's long-term success [29].

3.2.2.4 External Expertise

Look for external experts to help shepherd efforts, such as consultants, modernization specialists, etc. Their input can provide invaluable feedback to confirm that the project does adhere to standard development techniques [30]

Combining these strategies and best practices allows organizations to overcome some of the challenges of legacy application modernization. A properly designed and implemented modernization initiative can yield a more agile, streamlined, and competitive IT environment [32].

Figure 2: Challenges in Modernization of Legacy Application



In modernizing legacy applications, organizations face challenges that demand strategic thinking and adherence to best practices. To successfully navigate this complex terrain, it's crucial to begin with a comprehensive assessment of the existing legacy system, delving into its architecture, dependencies, data, and business processes. Clear requirements should be articulated, ensuring that all stakeholders are on the same page regarding the expected benefits of the modernization effort, thus preventing misinterpretations and scope creep. Effective change management using agile tactics is pivotal in overcoming end-user resistance and ensuring a smooth transition. Rigorous testing and validation are paramount, encompassing data migration, features, and integrations, to validate the system's compliance with service level objectives, security, and industry standards. Alongside these strategies, embracing best practices like meticulous documentation, incremental modernization, ongoing monitoring and maintenance, and seeking external expertise is imperative. This holistic approach empowers organizations to conquer the challenges of legacy application modernization, ultimately leading to a more agile, streamlined, and competitive IT landscape.

4. CONCLUSION

In conclusion, this paper explores different facets of legacy modernization and presents findings regarding its challenges, approaches, and recommended strategies. The major findings and key points can be summarized as follows:

a. Legacy Application Modernization Necessity: This study emphasizes the urgency of legacy application modernization, given its scalability, security, and integrative compatibility constraints with current innovations. While effective, these systems ultimately represent liabilities as the digital environment changes.

b. Drivers for Modernization: The research shows that legacy application modernization drivers come in pairs: the fast pace of business requires more agility/scalability/innovation at the same time as technology marches on, presenting both new opportunities/capabilities to be exploited and efficiencies which legacy systems do not harness effectively.

c. Challenges in Modernization: The study provides detailed insights into the various pain points faced during migration, including the dilemma between data migration and application migration, the complexity of legacy data stacks, user acceptance problems, seamless integration challenges, selection of migration strategy, and data migration obstacles.

d. Modernization Strategies: The research provides a detailed view of the modernization strategies, presenting organizations with various options—from encapsulation (or cocoons), rehosting, re-platforming, refactoring, and re-architecting to hybrid approaches. It highlights the need to align these strategies with organizational objectives and conditions.

e. Best Practices for Success: This comprehensive study reveals key tenets that are critical to ensuring a successful legacy application modernization, among which include performing thorough assessments, establishing clear requirements with well-defined KPIs, implementing an effective change management strategy, conducting rigorous testing and validation and thorough documentation.

To summarize, this research serves as an indispensable guide for companies embarking on the transformation journey of their legacy systems. It offers a comprehensive framework and

blueprint to navigate the challenges, make decisive actions, and ultimately transition from traditional methods to more adaptable, agile, and competitive digital alternatives.

REFERENCES

[1] AWS. (2021). Best Practices for Modernizing Legacy Applications. Are we speaking the industry language? The practice and literature of modernizing legacy systems with microservices. In *15th Brazilian Symposium on Software Components, Architectures, and Reuse* (pp. 61-70).

[2] Beebe, N. H. (2023). A Complete Bibliography of ACM Queue: Tomorrow's computing Today. doi:10.4204/eptcs.163.1

[3] Brinkmann, S., et al. (2017). Challenges in Modernizing Legacy Systems to Microservices Architecture. In 2017 IEEE International Conference on Software Architecture (ICSA).

[4] Carpenter, D. S. (2003). Legacy Modernization: Strategies for Modernizing Systems and Applications. Auerbach Publications.

[5] Clements, P. C., Kazman, R., & Klein, M. H. (2015). Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices. Addison-Wesley Professional.

[6] Debusmann, M., Schmid, M., & Kroeger, R. (2005, June). Model-driven self-management of legacy applications. In *IFIP International Conference on Distributed Applications and Interoperable Systems* (pp. 56-67). Berlin, Heidelberg: Springer Berlin Heidelberg.. doi:10.1007/11498094_6

[7] Ekanem, B. A., & Woherem, E. (2016). Legacy components stability assessment and ranking using software maturity index. *International Journal of Computer Applications*, *134*(13), 22-30. doi:10.5120/ijca2016908157

[8] Gartner, Inc. (2018). Strategies for Modernizing Legacy Applications. Gartner, Inc. Research Note.

[9] Gupta, A. A., & Vaishnavi, V. (2014). Legacy Systems: Transformation Strategies. CRC Press

[10] Hamza, H. S., et al. (2019). A Framework for Assessing Legacy System Modernization Challenges. In 2019 21st International Conference on Advanced Communication Technology (ICACT).

[11] Ilechko, P. (2005). Modernizing Legacy Applications: Designing, Coding, and Testing Java Applications. Addison-Wesley Professional.

[12] Infosys Ltd. (2020). Modernizing Legacy Applications in the Cloud Era. Whitepaper.

[13] Jones, P. M. (2016). Modernizing Legacy Applications in PHP. Leanpub.

[14] Khan, M., Ali, I., Nisar, W., Saleem, M. Q., Ahmed, A. S., Elamin, H. E., ... & Shafiq, M. (2022). Modernization Framework to Enhance the Security of Legacy Information Systems. *Intelligent Automation & Soft Computing*, *32*(1) doi:10.32604/iasc.2022.016120

[15] Kiran Mallidi, R., Sharma, M., & Singh, J. (2021). Legacy Digital Transformation: TCO and ROI Analysis. *International journal of electrical and computer engineering systems*, *12*(3), 163-170. doi:10.32985/ijeces.12.3.5

[16] Koegel, M., et al. (2016). Legacy Systems Challenges and Opportunities: A Case Study in Application Modernization. Procedia Computer Science.

[17] Kumar, A., & Garg, H. (2018). Modernizing Legacy Systems with Microservices and APIs. O'Reilly Media.

[18] Mainetti, L., Paiano, R., &Pandurino, A. (2012). Migros: a model-driven transformation approach of the user experience of legacy applications. In *Web Engineering: 12th International Conference, ICWE 2012, Berlin, Germany, July 23-27, 2012. Proceedings 12* (pp. 490-493). Springer Berlin Heidelberg. doi:10.1007/978-3-642-31753-8_51

[19] Mayor, J., Gomez, P., Chang, F., & Lupyan, G. (2014). Connectionism coming of age: legacy and future challenges. *Frontiers in psychology*, *5*, 187. doi:10.3389/fpsyg.2014.00187

[20] Microsoft Corporation. (2020). Modernizing Legacy Applications: A Microsoft Approach. Microsoft Corporation. Whitepaper.

[21] Pradhan, A. (2020). Legacy Modernization: Transforming Your Legacy Systems to the Modern Age. Apress.

[22] Radhakrishnan, H., Rouson, D. W., Morris, K., Shende, S., &Kassinos, S. C. (2015). Using coarrays to parallelize legacy Fortran applications: Strategy and case study. *Scientific Programming*, *2015*, 2-2.doi:10.1155/2015/904983

[23] Rodríguez-Echeverría, R., Conejero, J. M., Clemente, P. J., Preciado, J. C., & Sánchez-Figueroa, F. (2012). Modernization of legacy web applications into rich internet applications. In *Current Trends in Web Engineering: Workshops, Doctoral Symposium, and Tutorials, Held at ICWE 2011, Paphos, Cyprus, June 20-21, 2011. Revised Selected Papers 11* (pp. 236-250). Springer Berlin Heidelberg. doi:10.1007/978-3-642-27997-3_24

[24] Rodríguez-Echeverría, R., Conejero, J. M., Linaje, M., Preciado, J. C., & Sánchez-Figueroa, F. (2010). Re-engineering legacy web applications into rich internet applications. In *Web Engineering: 10th International Conference, ICWE 2010, Vienna Austria, July 5-9, 2010. Proceedings 10* (pp. 189-203). Springer Berlin Heidelberg, 189–203. doi:10.1007/978-3-642-13911-6_13

[25] Satyanarayanan, M., Eiszler, T., Harkes, J., Turki, H., & Feng, Z. (2020). Edge computing for legacy applications. *IEEE Pervasive Computing*, *19*(4), 19-28. doi:10.1109/mprv.2020.3026229

[26] Savio, P., Scionti, A., Vitali, G., Viviani, P., Vercellino, C., Terzo, O., ... & Poli, F. (2023) Accelerating legacy applications with spatial computing devices. *The Journal of Supercomputing*, *79*(7), 7461-7483.doi:10.1007/s11227-022-04925-2

[27] Sharma, R., et al. (2017). Modernizing Legacy Systems: Leveraging Microservices and DevOps. IBM Redbooks.

[28] Smith, J. D., & Brown, A. B. (2019). Legacy Application Modernization: A Comprehensive Framework. Journal of Software Engineering and Applications, 12(1), 12-29.

[29] Thompson, E. L., & Patel, S. R. (2017). Legacy Application Modernization: A Case Study of Industry Best Practices. International Journal of Software Engineering & Applications, 8(3), 129-141.

[30] Turner, R. K., & Davis, P. J. (2015). Legacy Application Modernization: Challenges and Solutions. In Proceedings of the International Conference on Software Engineering (ICSE), 2015, 145-154.

[31] Ulrich, W. M., & Newcomb, P. H. (2007). Legacy Modernization: A Pragmatic Approach. Cutter Consortium.

[32] Van Kranenburg, H. J., & Romeijnders, J. W. (2011). Challenges and Strategies for Modernizing Legacy Systems. International Journal of Computer Applications.

Acronyms and abbreviations used in the articles:

1. **CI/CD:** Continuous Integration/Continuous Deployment — a set of practices of software development aiming at the automation of building, testing and delivery of applications.
2. **DevOps:** A philosophy for developing and operating software that emphasizes communications, collaborations, integrations, and automations b/w the software's developer's and ITs operation's personnel's to streamline the development and deployment of high-quality software.
3. **SLO:** Service Level Objective — A set of criteria that describe the level of service quality that the user or client expects.
4. **IT:** "I have been in the same position, but that is no excuse for being rude."

APPENDIX

Search Table - 1

Database	Keywords	Search Results
SQLite	Legacy system modernization challenges	45

PsycINFO	User acceptance, change management	32
Scopus	Data migration, integration challenges	50
Web of Science	Modernization strategies, best practices	60

Literature Review Table - 2

Study	Objective	Findings
[1] AWS (2021)	Explore best practices for modernizing legacy applications.	Explores the modernization of legacy systems using micro services, providing insights and recommendations of the industry practices and literature.
[2] Beebe (2023)	Provide a bibliography of ACM Queue regarding computing trends.	N/A
[3] Brinkmann et al. (2017)	Examine challenges in modernizing legacy systems to micro services architecture.	Identifies challenges and potential solutions in transitioning from legacy systems to micro services architecture.
[4] Carpenter (2003)	Discuss strategies for modernizing legacy systems and applications.	Offers strategies and approaches for modernization, emphasizing the importance of legacy system transformation.
[5] Clements et al. (2015)	Explore modernizing legacy systems from a software and business perspective.	These include elements of software technologies; engineering; and, business practices, all geared the state of the art.
[6] Debusmann et al. (2005)	Development of a Model-Driven Self-Management Strategy to Manage Legacy Applications.	Discusses a model-driven approach to self-managing legacy applications for improved efficiency and maintenance.
[7] Ekanem &Woherem (2016)	Assess the stability of legacy components using a software maturity index.	Proposes a software maturity index for evaluating the stability of legacy components in modernization efforts.
[8] Gartner, Inc. (2018)	Provide strategies for modernizing legacy applications.	Offers insights and recommendations from Gartner on modernization strategies to meet evolving business needs.
[9] Gupta & Vaishnavi (2014)	Examine transformation strategies for legacy systems.	Discusses strategies for transforming legacy systems to align with contemporary technology and business requirements.
[10] Hamza et al. (2019)	Present a framework for assessing legacy system	Introduces a framework for evaluating challenges in legacy system

	modernization challenges.	modernization, aiding in decision-making.
[11] Ilechko (2005)	Focus on designing, coding, and testing Java applications for modernization.	Discusses the design, coding, and testing aspects of modernizing legacy applications using Java.
[12] Infosys Ltd. (2020)	Explore modernizing legacy applications in the cloud era.	Provides insights into modernization strategies and considerations specific to cloud-based environments.
[13] Jones (2016)	Discuss modernizing legacy applications in PHP.	Offers guidance on modernizing legacy applications using PHP, a widely-used scripting language.
[14] Khan, 2022)	Build a modernization framework for securing legacy information systems	It presents a model for making legacy applications secure during transformation.
[15] Kiran-Mallidi's et al., 2021)	Legacies digitals transformations TCOs/ ROIs analysis/ Calculate TCO/ROI for user legacy digital transformation.	Examines the financial aspects of legacy system modernization, focusing on TCO and ROI analysis.
[16] Koegel et al. (2016)	Explore challenges and opportunities in legacy systems with a case study in application modernization.	Presents a case study on application modernization, highlighting challenges and opportunities in legacy systems.
[17] Kumar & Garg (2018)	Discuss modernizing legacy systems with microservices and APIs.	Explores the role of microservices and APIs in modernizing legacy systems, emphasizing their importance.
[18] Mainetti et al. (2012)	Model-Driven UX Improvement in Legacy Systems.	Enhancing user experience in legacy applications by using a model-driven methodology.
[19] Mayor et al. (2014)	Explore connectionism and its relevance in modernization challenges.	How connectionism is important in solving the challenges of modernity.
[20] Microsoft Corporation (2020)	Provide insights into Microsoft's approach to modernizing legacy applications.	Presents Microsoft's approach to modernization, highlighting strategies and considerations.
[21] Pradhan (2020)	Focus on legacy modernization and transformation strategies.	Discusses various strategies for modernizing legacy systems, providing guidance for transformation efforts.
[22] Radhakrishnan et al. (2015)	Introducing Coarrays in the Parallelization of Legacy Fortran Applications: A Case Study.	Presents a case study on parallelizing legacy Fortran applications using coarrays, addressing modernization challenges.
[23] Rodríguez-Echeverría et al. (2012)	Modernize old e-commerce websites into rich internet applications.	Covers the conversion of traditional web apps to RIA — the way forward.

[24] Rodríguez, et al. (2010)	Talk about restructuring old-world websites as rich internet apps.	Transitioning Legacy Web Apps Into Rich Internet Application.
[25] Satyanarayanan, et al. (2020)	Investigate edge computing for heritage apps.	Talks on how edge computing can be applied to existing applications and overcomes modernization problems.
[26] Savio et al. (2023)	Accelerate legacy applications with spatial computing devices.	Presents the use of spatial computing devices to accelerate legacy applications, focusing on modernization.
[27] Sharma et al. (2017)	Discuss modernizing legacy systems using microservices and DevOps.	Explores the role of microservices and DevOps in modernizing legacy systems and provides insights.
[28] Smith & Brown (2019)	Present a comprehensive framework for legacy application modernization.	Introduces a comprehensive framework for modernizing legacy applications, aiding in the process.
[29] Thompson & Patel (2017)	Explore industry best practices in legacy application modernization through a case study.	Presents best practices in legacy application modernization based on a case study, offering insights.
[30] Turner & Davis (2015)	Discuss challenges and solutions in legacy application modernization.	Explores challenges and potential solutions in modernizing legacy applications, addressing common issues.
[31] Ulrich & Newcomb (2007)	Present a pragmatic approach to legacy modernization.	Introduces a pragmatic approach to modernizing legacy systems, focusing on practicality.
[32] Van Kranenburg & Romeijnders (2011)	Explore challenges and strategies in modernizing legacy systems.	Discusses challenges and strategies for modernizing legacy systems, providing valuable insights.

CASP Table - 3

Study	Was the study valid? (Yes/No)	What are the results?	Are the results applicable to the local population?
[1] AWS (2021)	Yes	Industry insights and	Applicability may vary

		recommendations for modernizing legacy systems with microservices.	based on local technology landscape.
[2] Beebe (2023)	Not applicable	N/A	Not applicable.
[3] Brinkmann et al. (2017)	Yes	Identified challenges and potential solutions in transitioning to microservices architecture.	Applicability depends on local adoption of microservices.
[4] Carpenter (2003)	Yes	Strategies for modernizing legacy systems and applications.	Applicable to local systems undergoing modernization.
[5] Clements et al. (2015)	Yes	Insights into modernization from software and business perspectives.	Applicability depends on local software and business context.
[6] Debusmann et al. (2005)	Yes	Using a model–driven approach in managing legacy applications.	Applicability depends on local use of model-driven techniques.
[7] Ekanem &Woherem (2016)	Yes	Proposed software maturity index for assessing legacy component stability.	Applicable to local assessment of legacy components.
[8] Gartner, Inc. (2018)	Yes	Insights and recommendations for modernization strategies.	Applicability depends on local business and technology context.
[9] Gupta & Vaishnavi (2014)	Yes	Strategies for transforming legacy systems.	Applicability varies based on local transformation needs.
[10] Hamza et al. (2019)	Yes	Framework for assessing modernization challenges.	Applicable to local assessment of modernization challenges.
[11] Ilechko (2005)	Yes	Guidance on designing, coding, and testing Java applications.	Applicable to local Java application modernization.
[12] Infosys Ltd. (2020)	Yes	Insights into modernization in the cloud era.	Applicability depends on local cloud adoption.
[13] Jones (2016)	Yes	Guidance on modernization in PHP.	Applicable to local PHP-based modernization efforts.
[14] Khan et al. (2022)	Yes	Framework for enhancing legacy system security during modernization.	Applicable to local security-focused modernization.
[15] Kiran-Mallidi's et al., 2021)	Yes	TCOs and ROIs analyzing in legacy digital transformation.	Applicability depends on local financial

			considerations.
[16] Koegel et al. (2016)	Yes	Case study on challenges and opportunities in application modernization.	Applicable to local cases of application modernization.
[17] Kumar & Garg (2018)	Yes	Exploration of modernization with microservices and APIs.	Applicability depends on local adoption of microservices and APIs.
[18] Mainetti et al. (2012)	Yes	Model-driven transformation approach for enhancing user experience.	Applicability depends on local use of model-driven techniques.
[19] Mayor et al. (2014)	Yes	Discussion of connectionism in modernization challenges.	Applicability varies based on local adoption of connectionism.
[20] Microsoft Corporation (2020)	Yes	Insights into Microsoft's modernization approach.	Applicability depends on local use of Microsoft technologies.
[21] Pradhan (2020)	Yes	Discussion of various modernization strategies.	Applicability varies based on local modernization needs.
[22] Radhakrishnan et al. (2015)	Yes	Case study on parallelizing legacy Fortran applications.	Applicable to local cases involving legacy Fortran applications.
[23] Rodríguez-Echeverría et al. (2012)	Yes	To explore the migration of web apps from traditional stateless client-server model towards full client-side rendered single-page apps.	Applicability depends on local web application context.
[24] Rodríguez, et al. (2010)	Yes	Legacy web page vs. the rich internet app debate.	Applicability depends on local web application context.
[25] Satyanarayanan, et al. (2020)	Yes	Exploration of edge computing for legacy applications.	Applicability depends on local edge computing adoption.
[26] Savio et al. (2023)	Yes	Acceleration of legacy applications with spatial computing devices.	Applicability depends on local use of spatial computing devices.
[27] Sharma et al. (2017)	Yes	Discussion of modernization with microservices and DevOps.	Applicability depends on local adoption of microservices and DevOps.
[28] Smith & Brown (2019)	Yes	Presentation of a comprehensive framework for legacy application modernization.	Applicable to local efforts involving comprehensive modernization

			frameworks.
[29] Thompson & Patel (2017)	Yes	Exploration of industry best practices in legacy application modernization through a case study.	Applicability varies based on local industry practices.
[30] Turner & Davis (2015)	Yes	Discussion of challenges and solutions in legacy application modernization.	Applicability depends on local challenges and solutions in modernization.
[31] Ulrich & Newcomb (2007)	Yes	Introduction of a pragmatic approach to legacy modernization.	Applicable to local pragmatic approaches to modernization.
[32] Van Kranenburg & Romeijnders (2011)	Yes	Exploration of challenges and strategies in modernizing legacy systems.	Applicability varies based on local modernization challenges and strategies.

Theme Table - 4

Theme	Studies Included
Modernization Strategies	[4] Carpenter (2003), [9] Gupta & Vaishnavi (2014), [11] Ilechko (2005), [13] Jones (2016), [20] Microsoft Corporation (2020), [21] Pradhan (2020), [27] Sharma et al. (2017), [28] Smith & Brown (2019), [30] Turner & Davis (2015),

	[31] Ulrich & Newcomb (2007)
Microservices and APIs	[1] AWS (2021), [3] Brinkmann et al. (2017), [17] Kumar & Garg (2018)
Legacy Component Assessment	[7] Ekanem &Woherem (2016)
Security Enhancement	[14] Khan et al. (2022)
Cost and ROI Analysis	[15] Kiran Mallidi et al. (2021)
Case Studies and Challenges	[2] Beebe (2023), [6] Debusmann et al. (2005), [16] Koegel et al. (2016), [22] Radhakrishnan et al. (2015), [23] Rodríguez, Echeverria (2012), [24] Rodríguez-Echeverría et al. (2010), [29] Thompson & Patel (2017)
Model-Driven Approaches	[6] Debusmann et al. (2005), [18] Mainetti et al. (2012)
Edge Computing for Legacy Applications	[25] Satyanarayanan et al. (2020)
Spatial Computing Devices	[26] Savio et al. (2023)