Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2022, 9(12):196-201**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Bridging the Gap: A DevOps Strategy for Modernizing Legacy Systems

Nagaraju Islavath

Independent Researcher
Email ID: islavath.nagaraju@gmail.com

Abstract Legacy systems pose serious problems regarding outdated technologies, scalability, and high maintenance costs in many organizations. Upgrading such systems will be one of the necessary steps for companies to be competitive and responsive to the demands of fast-moving markets. DevOps, a set of practices combining software development and IT operations-Ops-presents a powerful avenue for surmounting these challenges. The paper describes a complete DevOps strategy for modernizing legacy systems. DevOps will help the organization shift from legacy systems to modern architectures with automation, continuous integration, code delivery, infrastructure such as code, and containerization. The proposed strategy considers minimal downtime, better scalability, security improvements, and performance improvement. This paper's case studies and practical examples demonstrate how organizations can adopt DevOps to transform legacy systems with business continuity and improved operational efficiency.

Keywords DevOps, legacy systems, modernization, automation, CI/CD, containerization, Infrastructure as Code, IT transformation

Introduction

Legacy systems form part of the backbone of most organizations. Systems built many years, or even decades, ago are often unreliable; they involve outdated technologies that are difficult and time-consuming to maintain and integrate with newer systems. Many businesses have undergone digital transformation recently, and modernizing legacy systems has become increasingly realized. Unless kept up with, it generates technical debt, security vulnerabilities, and other operational inefficiencies that keep an organization from competing effectively in today's marketplace.

Modernization of a legacy system is intensely an undertaking that has unique challenges. Such systems are mostly tightly integrated, resting on close interdependencies, making the change process rather dicey and costly. Moreover, technical experience to keep them operational is a rare commodity gaining momentum as more engineers concentrate on modern technologies. Organizations that refuse to adapt risk falling behind the pace of others who can act quicker and more effectively since their modern, flexible IT architecture allows it.

DevOps is increasingly viewed as the key to unlocking this need for legacy modernization. It reduces the risk and increasing complexity of changing outdated systems through merging developments and operations, automated from much of what has been done by hand in the past. Emphasis on continuous integration, continuous delivery, and automated testing enables faster and more reliable deployment cycles, which is critical for modernizing fragile legacy environments. Also, DevOps helps in the creation of a culture of collaboration and shared responsibility that would overcome the silos in organizations, which contribute so much to barriers in modernization initiatives. It deals with how DevOps practices can modernize a legacy system. In this respect, it will discuss the most important challenges related to legacy systems, the advantages of introducing a DevOps approach, and some real-world cases of successful modernization. By the end of this paper, it should become

obvious that DevOps offers a pragmatic and efficient way for organizations in search of filling the gap between a legacy system and modern IT infrastructure.

Main Body

Problem Statement

The following are some challenges presented by legacy systems to modern organizations. One of the key concerns is that their maintenance cost is extremely high. As technology evolves, growth in support of the same becomes increasingly complex. The infrastructures they rest on usually require very specialized skills and hardware that are highly expensive to maintain; thus, operational costs are high and could be used for investment in innovation or newer technologies. Moreover, the un-scalability nature of the legacy system means an organization cannot scale up according to the growing number of users, thereby hurting its competitive performance.

Another major problem with legacy systems is the question of security. Because most of these systems were engineered before laying down modern security standards, they easily become susceptible to cyberattacks. Locating and patching vulnerabilities within such systems is tricky because their codebases are immensely complex plus poorly documented. Technologies they rely on are probably not supported by their original vendors, thus leaving organizations devoid of important security updates. This also increases the risk of leakage of sensitive information, which may have disastrous consequences in enterprises engaged in sectors like finance and health, where personal data protection is touchy.

The other biggest challenge is the impossibility of integrating legacy systems with modern technologies. Concerning this, during organizations' recent adoptions of cloud computing, AI, and machine learning, it has been found that their legacy systems cannot handle these advanced capabilities. Legacy systems are usually based on proprietary or outdated communication protocols that may not be easily connected to modern platforms. This lack of interoperability is not only a damper on innovation. Still, it creates silos of data, negating the possibility of making proper use of data either in decision-making or operational efficiency.

Other critical obstacles to digital transformation are the operational rigidities of legacy systems. In other words, modern IT environments are all about flexibility and speed, especially in companies embracing Agile development methodologies at an unprecedented rate. The monolithic architectures of legacy systems cannot easily be adapted to support iterative development cycles imposed by agile methods. This translates into longer development times, frequent downtime during updates, and unresponsiveness to changing business needs. This has been proved by the inability of organizations to rely upon legacy systems to maintain efficiency and competitiveness compared with other business organizations that have adopted more flexible and modern approaches.

Last but not least, there are the cultural and organizational issues with the legacy systems. Most organizations have legacy systems ingrained within their operational processes. Modernizing them thus requires a considerable level of change management. The modernization of such systems is usually resisted by both technical staff and business stakeholders since people are generally unwilling to let go of existing systems and procedures to which they are accustomed, even if they are outdated. Unless the strategy is well articulated to handle these human issues, modernization projects will either be significantly delayed or fail.

Solution: DevOps for Legacy Systems

A DevOps approach would substantially reduce most challenges of modernizing legacy systems. This is because one of the key benefits derived from DevOps is the emphasis on automation. Automating repetitive tasks, such as testing, deployment, and monitoring, reduces human error risks and makes the updates to legacy systems consistent and reliable. In particular, automated testing is crucial when operating with fragile legacy systems because, in this respect, developers can swiftly find emerging problems and fix them without impacting the whole system.

Continuous Integration and Continuous Delivery stand as core practices in DevOps that may prove very effective in the case of a legacy modernization project. CI/CD pipelines allow developers to continuously merge code changes into a common repository, where automated tests need to be performed for stability in the system. It helps the teams find the issues quite early in the initial stage of the development cycle, saving it from getting

into production with errors. CI/CD does this for legacy systems, and every change has a high risk of deploying updates more frequently and with many more degrees of confidence.

Another key DevOps practice that can be applied to integrating legacy systems is Infrastructure as Code. This process involves handling and setting up infrastructures with the use of coding rather than doing it manually. In this way, organizations can standardize their infrastructure configuration settings, ensuring that legacy systems are deployed using consistent and repetitive methods. IaC also facilitates the migration of legacy systems to cloud environments because infrastructure can be defined and managed by code without relying on erroneous manual configurations.

While working on renovating legacy systems, microservices and containerization are other approaches. Most of the legacy systems are still monolithic in architecture, wherein all the components are interlinked with each other tightly. This prevents updating or scaling any component independently of the rest in the system. Segmentation of such monolithic applications into smaller, independent microservices can easily facilitate modularity in legacy systems for organizations. And then there are the containers, courtesy of Docker or Kubernetes, further reinforcing this modularity by allowing each microservice to run in its isolated environment, a great way to deploy, scale, and update individual components without disrupting the whole system.

DevOps also fosters a culture of collaboration and shared responsibility that is essential to the success of any modernization effort. In most organizations, the development and operations teams work in a vacuum, more or less oblivious to the happenings of the other team. This creates enormous inefficiency and delay simply because one team often operates without regard for the other. DevOps enables them to work together through all phases of software development: a plan, development, deployment, and monitoring. Such collaboration will speed up team modernization and ensure a common understanding between development and operations teams about what the system can do.

Use Cases

Probably the most prominent example of DevOps usage for legacy system modernization is in the financial services industry. For example, an international bank of great size suffered from a very obsolete core banking system: cumbersome to maintain, with a complete inability to scale up and handle increasing customer demand. So far, a DevOps approach means that the bank can decompose a monolithic system into smaller microservices with an independently updated and scaled functionality. More or less, CI/CD pipelines laid the bedrock for this bank to deploy updates more often by reducing downtime and drastically improving the customer experience. Besides, the automation of some testing and deployment processes reduces further the risks of human error.

DevOps has also played a significant role in modernizing electronic health record systems within the healthcare industry. Most healthcare providers depend on old versions of EHR systems that were never designed to integrate with modern technologies such as telemedicine and AI-based diagnostic tools. With the implementation of a DevOps strategy, a leading healthcare provider has been able to modernize its EHR system and make it more scalable and easy to integrate with new technologies. The provider containerized the different components of the EHR system, making updates and maintenance much easier. This modernization effort has improved the efficiency of operations for this healthcare provider and enhanced patient care by allowing quicker access to critical medical information.

Other government entities adopted DevOps as part of their process to modernize various legacy systems. For example, one federal agency was tasked with handling public services until it began to experience strains on its older infrastructure, which could not support citizen demands for online services. It undertook a DevOps approach that ultimately migrated its legacy systems into the cloud since it was more scalable and secure. With IaC, the agency could standardize its infrastructure configurations so that new system setups could be done much faster, scaling into multiple regions with consistency and uniformity. This attempt at modernization enhanced the agency's marks in service delivery for its citizens and simultaneously reduced operational costs by doing away with the need for expensive, obsolete hardware.

Another use case can be found in manufacturing, where many companies are still running on older, legacy systems that control production lines and supply chains. A manufacturing company, with operations spread across the globe was at its wit's end trying to work with an antiquated ERP system that could not integrate with today's supply chain management toolsets. Then, they used DevOps to change their ERP, making it far more

flexible and open to integrations with new technologies. Performing automation and updates through CI/CD pipelines and deploying updates in the ERP system was now possible without interfering with production, thus reducing downtime.

The retail sector has implemented DevOps to update in-store POS systems, many built on older technologies. For example, one large retailer reported significant frustrations with an outdated POS system that could not handle the high volume of transactions demanded by peak shopping seasons. With a DevOps strategy, it attained scalability and reliability while it renewed its POS system. These microservices allowed the retailer to further break down the big monolithic POS system into small independent components that could be independently updated and scaled. This modernization effort improved the retailer's ability to handle transactions and enhanced customer experience, with checkout times reduced.

Impact

The advantages of applying DevOps to the older legacy systems are huge, from very short to very long-term ones. One of the immediate benefits can be a reduction in downtown. Most of these legacy systems are prone to going down frequently because they are complex and fragile simultaneously. Following the automation, CI/CD allows an organization to reduce the chance of going down during updates and maintenance. With Test Automation, the changes are made through exhaustive testing before deployment to production, avoiding unexpected bugs that lead to system outages.

This reduces the operational cost for an organization in the long run. The maintenance cost of legacy systems is very high, as they require specific expertise in certain areas and usually run on outdated hardware. With DevOps, the modernization of these systems means eliminating such expensive hardware and simultaneously reducing the need for specialized technical expertise. Automation further reduces the number of people required to maintain and upgrade the system, reducing operational costs.

The business advantages of DevOps' adoption for the modernization of legacies include making the organization more agile and competitive. By modernizing their legacy, companies can respond much faster to the demands of the market or customers, either of which may be driven by regulatory requirements. Microservices and containerization allow easier updating and scaling of single system parts towards faster release cycles and less time to market for new features and services. In return, this fosters even more agility, particularly for finance and healthcare, where the potential to adapt quickly to regulatory environments and customer needs is at stake.

DevOps has also influenced security a lot. Legacy systems are usually quite vulnerable to cyberattacks because of their outdated architecture and security features. Employing DevOps practices, such as automated security testing and continuous monitoring, allows them to enhance their legacy system security. Automated security testing is integrated into the CI/CD pipeline, ensuring the vulnerabilities are picked up and fixed before deploying the code into production. Continuous monitoring helps organizations identify security-related incidents to take the necessary remediation actions in real time, thus minimizing the risk of data breaches and other security incidents.

Notably, the cultural impact of adopting DevOps principles cannot be second-guessed. As earlier said, one of the core philosophies of DevOps is the interaction of development and operations. These two departments have always been categorized in most organizations, especially those reliant on legacy systems, leading to inefficiencies and communication gaps. DevOps endeavors to ensure good interaction toward a common objective, with collaboration and shared responsibility between two teams that have been kept apart for so long. This cultural shift promotes more efficient modernization and better outcomes, where both teams are aligned to achieve mutual goals of improved performance and system reliability.

Scope

The scope of the application of DevOps to legacy systems is huge and cuts across all sectors. While the examples above are Milanese-tailored, focusing on industries related to finance, healthcare, government, manufacturing, and retail, the approach of DevOps can be adapted to practically any industry dependent on legacy systems. One of DevOps's main strengths is the flexibility to mold it to meet certain organizational needs or challenges. Be it the modernization of a legacy ERP system, POS system, or core banking system, an organization can apply the principles of DevOps to smoothen the process of its modernization.

DevOps adoption is essential in the financial services industry, as many banks and financial institutions depend on legacy systems from which critical operations are run. The ability to modernize such systems while keeping those systems secure and in full compliance with regulations is key to the success of any financial organization. DevOps provides the tooling and practices necessary to ensure that legacy system updates are securely and reliably deployed with a minimum chance of downtime and security breaches. Another area where the scope of DevOps is very high for legacy modernization involves the healthcare industry. Several care providers worldwide still depend on their old legacy EHR systems, which are necessary to maintain patients' data and ensure continuity of care. Modernizing these systems through the practice of DevOps can improve patient care since the provider can get access to medical information faster without any hitches, reducing the possible occurrence of errors due to updates. Automation and application of continuous integration or delivery pipelines are essential in healthcare services where the consequences for patients may be fatal in case of downtime.

DevOps can be used in manufacturing industries to modernize legacy systems that are responsible for managing production lines and supply chains. Many manufacturers are now adopting new technologies like IoT and AI, among others, to improve operation efficiency, and therefore, there is much urgency to modernize legacy systems. DevOps provides a framework for integrating these new technologies with legacy systems so the manufacturer can leverage the latest innovations with minimum disruption to his operations.

Another domain where DevOps applies to legacy modernization is within government sectors. Most government agencies depend on very old IT systems for delivery in public service, and the capability of modernizing these systems is critical to the realization of efficient and accessible government services. DevOps will thus enable such government agencies to migrate their legacy systems to cloud-based infrastructures that are more scalable and secure. IaC also plays an important role in the government sector, where agencies can easily standardize their infrastructure configurations and ensure their new systems are deployed consistently and securely.

Again, the retail sector is a field where the scope of DevOps within legacy modernization is unlimited. Most retailers still rely on legacy POS systems to process transactions, and the capability to modernize such systems is essential for improved customer experience. DevOps provides the tooling and practices that ensure POS system updates are deployed quickly and reliably, reducing system downtime during peak shopping seasons and improving transaction processing in general.

Conclusion

One of the most important challenges organizations in all fields face is modernizing legacy systems. While reliable, these systems are often outdated and a headache to maintain, increasing operational costs, security vulnerabilities, and limited scalability. DevOps provides a strong solution to such challenges by enabling automation, continuous integration/continuous deployment, infrastructure as code, and containerization within the bounds of modernization (Wolfart et al.,). It can reduce downtime, improve security, and make the legacy system highly scalable for an organization that embraces DevOps to keep it competitive within the demanding market.

The impact of modernizing the legacy of DevOps is huge, both operationally and from the standpoint of business agility. An organization can use the adoption of DevOps to reduce operational costs, enhance its ability to respond to changes in market conditions, and thereby embed greater security within the core legacy systems themselves. Another key benefit of DevOps is that the cultural shift in collaboration and shared responsibility ties the gap between development and operations teams, leading to more productive and successful modernization projects.

DevOps represents a pragmatic approach and an effective strategy for renovating outdated legacy systems. Because of its flexibility and scalability, it can be extended from the financial and healthcare sectors down to the manufacturing and retail sectors. The line for digital transformation will include DevOps adoption for legacy modernization, which is crucial to ensure that the business keeps in step with the changing technologies and is competitive within an ever-evolving marketplace. This might be further explored in the future where emerging technologies such as AI and machine learning extend DevOps to give organizations more advantages in modernizing their IT infrastructure.

References

[1]. Adriano, D. M. (2021). *DevOps and Information Technology Service Management: A Problem Management Case Study* (Master's thesis, ISCTE-Instituto Universitario de Lisboa (Portugal)).

[2]. Al Masud, S. M. R., Masnun, M., Sultana, A., Sultana, A., Ahmed, F., & Begum, N. (2022). DevOps Enabled Agile: Combining Agile and DevOps Methodologies for Software Development. *International Journal of Advanced Computer Science and Applications*, *13*(11).

[3]. Jussila, T. (2021). DevOps in mobile game development: An action research on applying DevOps practices in a project.

[4]. Khabouze, R. (2022). *Modernization of Legacy Information Technology Systems* (Doctoral dissertation, Walden University).

[5]. Ogala, J. O. (2022). A Complete Guide to DevOps Best Practices. *International Journal of Computer Science and Information Security (IJCSIS)*, *20*(2).

[6]. Oliveira, T. C. D. S. (2022). *Barriers and enablers of digital transformation: an exploratory analysis of digital debt and devops* (Doctoral dissertation).

[7]. Wolfart, D., Assunção, W. K., da Silva, I. F., Domingos, D. C., Schmeing, E., Villaca, G. L. D., & Paza, D. D. N. (2021, June). Modernizing legacy systems with microservices: A roadmap. In *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering* (pp. 149-159).

[8]. Woods, E., Erder, M., & Pureur, P. (2021). *Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps*. Addison-Wesley Professional.