sciendo

Efficiency and Cost-Effectiveness in Agile DevOps with Cloud Computing

Marian STOICA
Academia de Studii Economice, București, România
marians@ase.ro

Alexandru Ionuț NIȚU*
Academia de Studii Economice, București, România
**Corresponding author, alexandru.nitu@csie.ase.ro*

Abstract. *This article explores how Agile methodology, DevOps principles, and Cloud Computing fit into the software industry. The synergy study focuses on Axes Software, the oldest logistics software company in Romania. The effects of the association of these technologies during the development cycle of a software product are explored. Both their impact in the operational activities and the transformation of the development mode, as well as the financial impact, are evaluated. In the first part, theoretical aspects of Agile and DevOps ways of working and cloud technologies are explored. The amplification of the potential of the association of the three concepts is examined. With the basis formed, the paper continues with a detailed case study of Axes Software. It looks at the adoption of new technologies and way of working. Challenges and successes encountered are also considered. Key performance indicators such as release frequency, bug incidence (dropped from 72 to 4), and development velocity are examined to validate the effectiveness of this integration. The financial aspects, including initial investment and savings, are also analysed, leading to an assessment of the return on investment. Notably, deployments of new software versions now take two days instead of eleven, while the market witnessed a 275% increase in clients preferring the company's cloud-hosted solutions over on-premises options. Finally, the effects of the association for software developers are distinguished, emphasizing continuous learning, leadership vision and supporting digital transformation.*

Keywords: Agile Methodologies, DevOps Practices, Cloud Computing, Software Development Efficiency, Operational Cost-Effectiveness, Digital Transformation, Continuous Learning

Funding: This paper was co-financed by The Bucharest University of Economic Studies during the PhD program

Introduction

The association of Agile methodologies, DevOps practices and Cloud Computing technologies is not a trend in the field of software development, but a necessity for the transformation and reconstruction of work procedures used in companies. Their association is studied in the present paper. Both the impact on the efficiency achieved in operational processes and the transformation of the way of working, as well as the financial implications, are investigated. Axes Software is a company from Romania, founded in 2005. It stood out in the logistics software development sector. The company's active concerns include new Cloud Computing technologies, agile way of working and improving collaboration between software development and operations. The results of the trio combination study on the impact it had on both the method and the output of software development production are discussed.

DOI: 10.2478/picbe-2024-0287

Agile, rooting back to the Agile Manifesto of 2001, has championed a flexible, evolutionary approach to software development (Bucena & Kirikova, 2017). At the same time, due to the ever-increasing need for continuous software development, collaboration and automation between development and operations, the strong link satisfied through DevOps has been created. The scalability, ability to scale resources as needed, and on-demand availability of cloud services have created a conducive environment in which the Agile methodology and DevOps work environment can thrive. Moreover, they have enabled companies to adapt much more quickly to the demands of the markets in which they operate and to technological advances. The paper also considers the cultural and strategic variations required as a result of implementing such a complex adoption, in addition to examining the implementation of new technologies. The Axes Software case study shows the results, challenges and journey of this complex transformation. The journey from initial investments in infrastructure and training to the realization of tangible and intangible benefits post-implementation is examined, offering a real-world perspective on the effectiveness of this integration.

Key performance indicators such as release frequency, bug frequency and development speed are part of the analysed data. Financial aspects are also considered, understanding the costs, revenues and profitability Axes Software has gained as a result of the transition. Next, the wider impact on the software development sector is discussed. Concretely, the impact includes the special importance of the accumulation of new theoretical knowledge, the imposition of a new way of collaboration and the obligation to involve the company's management in the transformation initiative. The results obtained support the adoption of these methodologies and technologies and can be a guide for other companies choosing this path of digital transformation.

How the Agile-DevOps-Cloud Computing trio is integrated to increase efficiency in order to achieve a faster, more profitable and more efficient software development process is analysed from a practical perspective. The case study demonstrates a promising future for those who choose to change and adapt to new ways of working in the software development sector, so the process is beset by challenges at every turn.

Literature review

This section surveys the current state of research on the integration between agile software development through DevOps and cloud computing. The theoretical basis and key concepts relevant to this study is constituted by the review of existing literature. It comprises academic papers, reports and practical case studies, providing a detailed insight into the issues researched.

As defined by Fitzgerald & Stol (2017), in today's rapidly shifting business landscape, software development must adapt to deliver faster and meet evolving customer demands. While agile practices have been widely adopted, they primarily focus on development, leaving a gap in the overall software process. To bridge this gap and provide increased added value, companies are integrating DevOps, as Azad (2022) notes in his paper. This emphasizes collaboration between development and operations teams. Releasing software at a faster pace and with higher quality is supported by DevOps. End-user feedback is also taken into consideration.

Collaboration, adaptation, and development through iterations are emphasized by agile methodologies. They have thus transformed the traditional way of software development. Agile methodologies, stemming from the principles outlined in the Agile Manifesto, and the Scrum framework prioritize iterative development, collaboration with customers, and continual enhancement (Sutherland, 2014). As per to Stoica et al. in their comparison between Agile and

traditional software development paper (2013), these foundational frameworks offer a methodical strategy for software development, accentuating flexibility, and the ability to promptly adjust to evolving needs, prioritizing flexibility, and consistent customer feedback. Agile companies understand the need to pay more attention to customer needs and embrace evolution. This leads to faster delivery of higher quality software. Without the adoption of these principles, the improvement of software development processes would not have been possible.

According to Marnewick and Langerman (2020), DevOps is an extension of Agile principles that promotes the integration of development and operations teams. Its goal is to streamline the software delivery process by reducing the number of manual interventions and the time required to publish code changes to production environments. Theoretical notions regarding integration and continuous development and principles of "DevOps culture" are integrated into the DevOps way of working, their essence being collaboration and automation. A similar perspective, namely that DevOps fosters a culture of collaboration, continuous integration, and continuous delivery (CI/CD), all of which align with Agile values, is supported in Klotins & Perez-Andersson's (2022) work.

Hyrynsalmi (2022) asserts that Cloud Computing, grounded in theoretical principles such as virtualization, elasticity, and service models (like IaaS, PaaS, SaaS), presents an innovative approach to provisioning infrastructure and deploying software. The theoretical notions in this field emphasize the scalability, flexibility and net income of cloud-based solutions that allow companies to adapt more quickly to market demands. Resource maximization is also considered.

The move to cloud computing exposes full advantages including scalability, as companies have the freedom to adapt the computing power consumed to operational needs. Cloud hosting service providers offer flexibility through the wide portfolio of services made available to customers. According to Tang et al. (2014), providers mostly accept consumption-based payment. Significant initial investment in hardware infrastructure is thus reduced. Cost efficiency is thus obtained, reducing the financial pressure of new projects. Moreover, the services offered by the largest providers are often on a global scale, which enables companies to have a global presence.

The convergence of cloud computing and Agile DevOps represents a powerful synergy with the potential to revolutionize software development and delivery in several profound ways as highlighted by Khan et al. (2017). In a study made by Sravan et al. (2023), the particular need of Agile DevOps teams to benefit in the shortest possible time from infrastructure made available for development and test environments is fully satisfied by cloud hosting service providers. This is one of the biggest benefits, as it reduces software solution delivery times and supports both iterative working and rapid experimentation. Another standout factor is rapid vertical and horizontal scaling, which enables companies to adapt their hardware resource requirements to their workload. In addition, CI/CD integration functionalities are also coming. All of this makes the cloud environment perfectly aligned with Agile principles. Cloud-based CI/CD tools and services modernise the development pipeline by facilitating parallel testing, accelerating builds, and providing scalability, all contributing to the swiftness and efficiency of software delivery as described by Srithar et. al (2022). Seamless collaboration between Agile teams is enabled by adopting cloud-based tools and instruments. Efficiency and an increase in the culture of collaboration are thus achieved, a basic condition of Agile principles. They bring the company closer to success in a fast-paced development environment.

The results of scientific research and case studies present the practical implications of the integration of the Agile-DevOps-Cloud Computing trio. They note quantifiable benefits such as reduced time to publish code changes, vertical and horizontal scalability, and better cost control.

Specifically, a study conducted by Nasreen Azad (2022), considers the most critical aspects on which the success of DevOps projects depends. The data is based on the perspectives of 15 Devops engineers. They exposed the indicators of success grouped into categories related to organization, society, culture and technology by analyzing the answers to open questions. The research aligns with the academic literature, despite the fact that the actual impact cannot be attributed exactly to the identified factors. Moreover, the significance of the factors grouped into the four categories is emphasized, especially their presence during the adoption of new practices and processes even by companies outside the software development field, which reveals their importance as success factors. The results of this study denote the importance of continuing research on this topic and encourage both an increase in the sample of interviewees and a qualitative approach, despite the fact that no new indicators were identified. The authors encourage researchers to focus on improving the model, especially the technical factors.

Khan et al. (2022) warn about the underestimation of the human factor in the process of integrating these methodologies and technologies. Poor communication, collaboration and expertise are often at the root of challenges in this process, despite a strong desire to adopt state-of-the-art technologies. The presence of the mentioned barriers draw attention to the consideration of the human factor alongside technological changes. Diel et al. (2016) further elaborate on this notion, emphasizing that the success of integrating Agile, DevOps, and Cloud Computing hinges not solely on technological aspects, but substantially on human factors. These include alternations in organizational culture, resistance to change due to new methodologies, and reluctance to abandon the usual way of working. Clivanti (2021) and Amazon (2022 describe the high value of developing a culture of learning and transparency. They give as a model in this respect the companies Etsy and Capital One.

The vast field of software development requires understanding and familiarity with the distinct but interconnected domains of Agile methodologies, DevOps practices, and Cloud Computing technologies. These three contribute practices, principles and benefits as well as challenges. Next, the table below presents a comparative overview of these methodologies and how they work for better understanding. The contrast between Agile, DevOps and Cloud Computing is highlighted. The underlying principles, advantages and challenges are highlighted. I thus appreciate the synergy of the methodologies and the ability to complement and improve each other. Their influence in the development of modern software applications when integrated should also be noted.

Table 1. Comparative Overview of Agile, DevOps, and Cloud Computing Methodologies

Attributes	Agile	DevOps	Cloud Computing
Core Principles	• Adaptive planning • Early delivery • Continuous improvement • Flexibility	• Collaboration between dev and ops • Automation • Continuous integration and delivery • Monitoring and feedback	• On-demand resources provisioning • Scalability • Broad network access • Measured service
Key Practices	• Iterative development • Regular feedback • Cross-functional teams	• CI/CD pipelines • Automated testing • Infrastructure as Code	• Virtualization • Resource pooling • Self-service portals • Automated scaling
Tools & Technologies	• Scrum boards • Kanban boards • Agile project management tools	• Configuration management tools • Deployment automation tools • Monitoring an analytics tool	• Virtual machines • Containers • Cloud storage • Cloud management platforms

Attributes	Agile	DevOps	Cloud Computing
Benefits	• Faster response to change • Improved customer satisfaction • Increased team morale	• Faster deployment cycles • Reduced silos between teams • Improved deployment quality	• Cost efficiency • High accessibility and flexibility • Improved disaster recovery
Challenges	• Scope creep • Overemphasis on frequent delivery • Resistance to change	• Complexity in implementation • Need for cultural change • Security challenges	• Data security concerns • Dependency on service providers • Compliance and legal issues

Sources: (Sutherland, 2014; Stoica, Ghilic-Micu, & Mircea, 2013; Bucena & Kirikova, 2017; Srithar, et al., Cost-Effective Integration and Deployment of Enterprise Application Using Azure Cloud Devops, 2022; Tang, Lee, & He, 2014; Bucena & Kirikova, 2017)

The preceding tabular presentation provides a comparison of the elements under discussion and their contribution to software development.

The review of both trio and individual interactions have been analysed in this section from the point of view of the existing literature. This sets the stage for the following sections that will detail how these methodologies and technologies integrate to produce efficiency and profit.

Methodology

Due to the ever-changing software development techniques, a review of methodologies that emphasize flexibility, collaboration, and rapid production is required. In what follows, the paper focuses on the Agile-DevOps-Cloud Computing synergy, the goal being to promote an ideal working environment for software development, its rapid and efficient deployment and scaling.

Agile, DevOps, and Cloud Computing intersect at Axes Software's R&D Department, which utilizes Azure DevOps for version control and development, following Agile principles through stand-ups, sprints, and reviews. Cloud hosting of applications enables CI/CD adoption. This facilitates rapid integration of changes and thus streamlines development. Agility, scalability, and efficiency are therefore ensured.

Axes Software has set goals for transforming the software development process. They stem from the desire to provide customers with superior software. At the same time, the optimization of internal resources is also desired. The foremost aim is to upgrade source code management, foster stronger collaboration among development teams, and implement the changes to software with minimal errors. To achieve this, Axes Software decided to implement a version management software system that supports the DevOps way of working and practices.

Bringing Agile, DevOps and Cloud Computing together brings significant benefits to companies in terms of development costs and efficiency. Traditional financial models that rely heavily on capital spending undergo a paradigm shift when businesses transition to the cloud. Instead of incurring upfront costs on infrastructure, organizations can transition to an operational expenditure model, provisioning resources only when necessary and scaling back when demand subsides. This financial flexibility is further complemented by the cloud's inherent ability to eliminate overhead costs, such as hardware maintenance, space allocation, and energy consumption (Wang, et al., 2010).

In addition to the promising benefits of integration, the challenges that arise from this association must also be considered. One of the most common is that of data security. The network of servers and interconnected equipment that is made available by cloud hosting service providers

can be perceived as a data security risk. The risk increases especially in multi-tenancy work environments. As Agile and DevOps practices emphasize speed and continuous delivery, it becomes imperative to intertwine rigorous security measures throughout the development lifecycle to mitigate potential risks (Nitu, 2022). Another challenge that organizations often struggle with is vendor lock-in. Becoming heavily reliant on the tools and services of a single cloud provider can restrict flexibility and make future migrations or integrations frightening and potentially costly. In addition to the concerns highlighted up to this point, the learning curve and adoption of these technologies must also be considered. Continuous professional training through workshops and the improvement of company employees become indispensable. The success of exploiting the full potential made available by new technologies depends on them.

Significant challenges were also encountered in the stage of creating database projects to benefit from CI/CD. These had as predominant causes the lack of experience and the undervalued effort. In addition, on-premises hosting of customer applications due to personal choices was another obstacle. Although significant, each of these obstacles has been overcome, highlighting the flexibility and adaptability of DevOps practices and tools.

Case Study: Embracing Agile, DevOps, and Cloud at Axes Software - A CTO's Perspective

Axes Software, a name synonymous with cutting-edge technological solutions, stands as a testament to this transformation. As one of the industry's leading players in software for logistics in Romania, founded in 2005, the journey of integrating these paradigms has not been merely about adopting the latest tools or methodologies. It was actually a strategic decision based on the desire to optimize the software development process, increase internal collaboration and deliver scalable products as a result of market demand. At the crossroads of this transformation has been the role of the Chief Technology Officer. Through this lens, I've witnessed, influenced, and often navigated the challenges and opportunities presented by this integration. The decision to anchor our efforts on Microsoft's Azure DevOps and Visual Studio was not taken lightly. The decision to use the latest tools was a clear and assumed one. The company's principles and technological aspirations were considered.

The transformation process required to adopt Agile, DevOps and cloud computing methodologies and technologies was undertaken by Axes Software company management. The challenges and key actions that were part of this project are also detailed. At the heart of this process is Azure DevOps. It goes far beyond the framework of a simple tool by seamlessly integrating Agile principles and DevOps methodology. With the help of this tool, the software development process has been refined starting from changing the way of backlog management to CI/CD. Also, native code development was chosen for cloud hosting to take advantage of all the facilities offered by cloud hosting. Additionally, Visual Studio not only serves as a coding environment but also complements our Agile principles. Its harmonious integration with Azure DevOps amplifies software delivery efficiency. However, this transition encountered hurdles. It's noteworthy that while tools like Azure DevOps and Visual Studio act as catalysts, the real metamorphosis at Axes Software involves amalgamating these tools with a culture tuned to Agile and DevOps. Based on the experience gained, we could not help but notice the substantial benefits of combining new technologies, adapting processes and improving culture.

At the heart of our company's software ecosystem lies a suite of applications that might be described as layered classics. These applications, predominantly desktop-based, are constructed on

a three-tier architecture. User interface, business logic and data access work together within the applications to provide complex and reliable solutions that meet the demands of our customers. The primary technology stack involves C# for the desktop components, ensuring rapid, efficient, and scalable application development. On the database front, T-SQL plays an important role It successfully manages complex and critical customer data. It also provides the necessary flexibility in working efficiently with large amounts of data. This is the moment where our approach took a slightly unconventional turn. While using Visual Studio for C# development is standard, leveraging it for SQL is less common. It raised eyebrows initially. The development community often perceives Visual Studio as an environment tailored more for application development rather than database scripting.

Out of the desire to improve the software development process, we made the decision to introduce source code control through versioning and code review. These additional steps have been introduced into the standard workflow. Among the expected benefits are decreased implementation times by introducing CI/CD and improved coordination between development, implementation and customer support teams. These were preceded, however, by delays in the development process mainly caused by the introduction of rigorous validation. Once the teams adapted to the new way of working, efficiency was significantly improved. By automatically triggering the code review process when release requests are created, the number of potential errors in production has decreased significantly. Overcoming these obstacles required a sustained effort to convince colleagues to adopt the new way of working. This consisted in particular of giving increased importance to tests and validations. In addition, programming directly in test and production environments was no longer possible, a habit deeply ingrained in the way of working. Eventually, the mindset was successfully changed, leading to a controlled way of working.

As a result of the transformation, we noticed a significant decrease in the number of bugs. Thus, the reliability of the software has increased. Through this process we also achieved a versioning of the sources which in turn led to full change auditing possibilities, previously unavailable.

In the ever-evolving software development industry, Axes Software's decade-long reliance on Tortoise SVN proved advantageous. Technological advancement has widened the gap between the tools used by the company and those available in the market. At the base, there was the desire to identify a modern work tool. Following a complex analysis that included platform versatility and anticipated needs, we chose Azure DevOps. Its adoption began through training sessions. Once we recognized its potential, we meticulously migrated projects from Tortoise SVN, ensuring the preservation of legacy data and workflows. This transition was accompanied by thorough in-house training to familiarise the team with the workings of the new platform.

CI/CD is a methodological approach that emphasizes the regular integration of code changes into a central repository, followed by automated testing and deployment. This process ensures that software development is more streamlined, efficient, and less prone to errors, significantly enhancing the speed and quality of software delivery. Fig. 1 visually outlines the CI/CD pipeline, showcasing the sequence of stages a software artifact across various servers, each tailored for a different customer. Each customer's server is equipped with an Azure DevOps agent. The deployment process is conducted through these agents, effectively demonstrating how the CI/CD pipeline triggers the deployment of the software artifact. This setup ensures that each update or new release is seamlessly and efficiently rolled out across the diverse servers, maintaining consistency and reliability in the delivery to each customer.

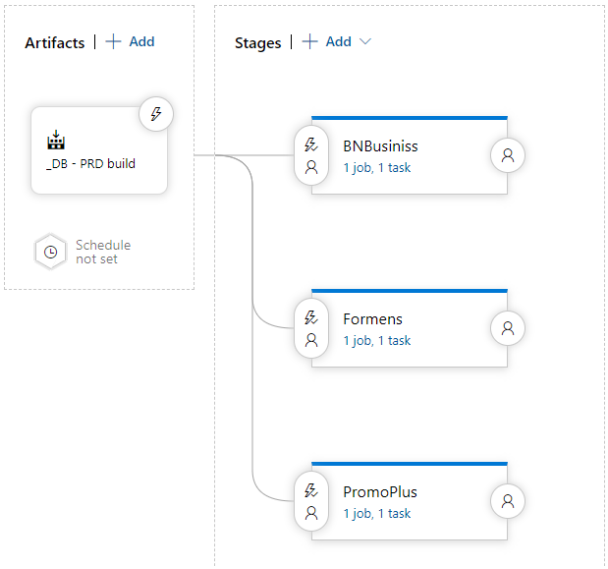

Figure 1. CI/CD Pipeline for Database

Source: dev.Azure.com.

Migrating to Azure DevOps introduced us to an elaborate world of branching. For us, the "development branch" serves as the starting point for each feature, subsequently progressing to testing, and finally culminating in production. This branching structure ensures a streamlined flow, but implementing it wasn't straightforward. Fig. 2 offers a glimpse into the collaborative review process, showcasing a real example of code submission within Azure DevOps environment. This specific submission, instigated by a developer, illustrates the fundamental components of code review and approval protocols.

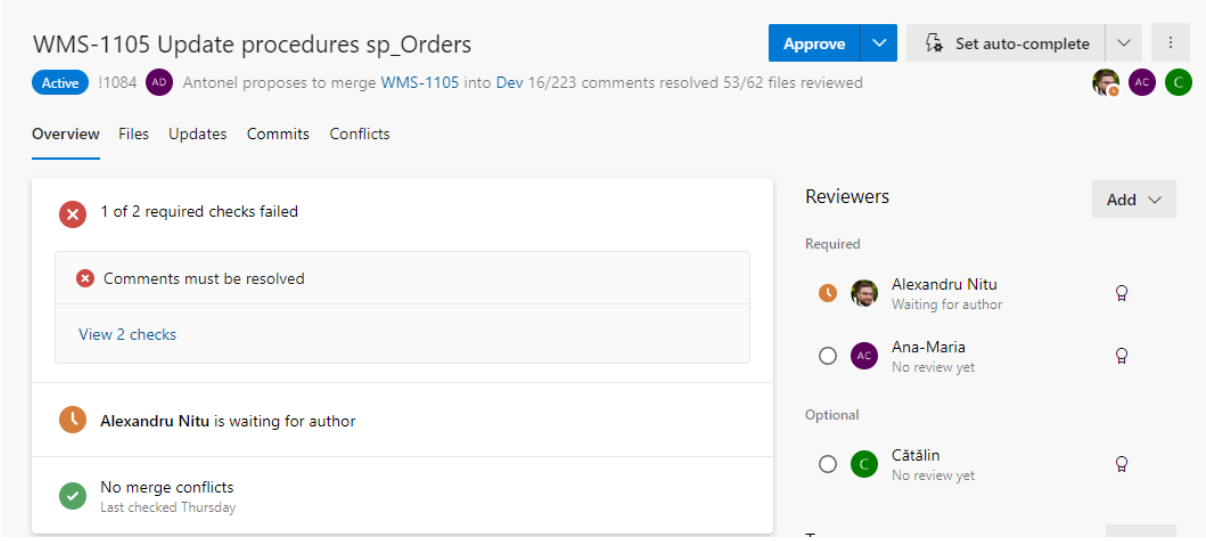

Figure 2. Pull Request Review in Azure DevOps

In the image, you will notice that the pull request contains a significant update, encompassing 62 files. I have implemented a detailed and rigorous code review system to improve the quality of changes as they are published to existing code. My role, along with a colleague from the Quality Assurance (QA) team, is highlighted as required reviewers, ensuring that any changes

meet our stringent quality standards. Additionally, another colleague is listed as an optional reviewer, providing an opportunity for broader team input and knowledge sharing. A notable aspect of this pull request is the extensive feedback given, with 223 comments made, of which 16 have been addressed so far. Substantial involvement in the code review process is necessary to maintain a high quality of the source code. It also contributes to compliance with the project requirements, being an additional check. For this pull request to progress to the next stage, namely merging into the development branch, two key conditions must be met: all comments and feedback must be resolved, and we must receive approval from the QA department. Additionally, the platform performs its own checks as part of what we refer to as the "build policy"' It verifies if there are any merge conflicts and confirms that the project compiles correctly without errors. We achieved an additional level of validation that strengthened the reliability of our implementation process.

The transition from legacy versioning and control software to Azure DevOps has yielded significant benefits. Important objectives were thus achieved, such as decreasing the number of reported errors and improving the software development process. With this change, it became possible for several programmers to work on the same projects at the same time. The previously existing blockages were thus removed. This change has enabled us to achieve a high standard for the entire portfolio of 13 applications. This resulted in a simplified workflow with fewer errors.

In our company, the journey towards DevOps adoption began not with a grand mandate from the top, but rather, organically, from the very core of our technical operations - the R&D department. This wasn't a decision about choosing a new set of tools; it was a transformation in the very way we approached software development, deployment, and maintenance. Previously, using Visual Studio for writing SQL code was a non-starter. However, adopting the DevOps methodology includes not only changing the tools used, but also changing the processes and culture of the organization. With DevOps, the idea was to break down silos, encourage collaboration, and bring about a seamless flow of work from development to deployment. While the R&D department was affected by the transition, other departments also began to be influenced. Implementation, once seen as a separate domain, began to integrate DevOps principles. What was once a segmented, step-by-step procedure transformed into a more fluid and interconnected process.

Transparency and accountability in source code management were introduced with these changes. These can be considered the most important benefits. Gone were the days when changes in the source code were shrouded in mystery. With the new DevOps practices in place, the Support team had a clear view of the modifications. Every change could be traced back to its origin - who made the change, when was it done, and most importantly, why it was made. The sense of personal responsibility was thus stimulated, and the number of errors was reduced. Everyone knew that their contributions were visible and would be held to the highest standards.

The presented process was a complex one, resulting in remarkable improvements in the entire software development process. Regular feedback loops, the introduction of a rigorous dual-check system, and a refined branching methodology modelled after successful GIT branching models (Chang, Ou, & Deng, 2019) were important in this transformation. As the company continues to grow, the desire to provide quality software products remains essential. This will continue to be supported by the extended capabilities of Azure DevOps.

Results and Discussions

The previous section outlined how Axes Software embraced the trinity of Agile, DevOps, and Cloud Computing. In what follows, a Chief Technology Officer's (CTO) view of the practical and

strategic changes to move toward a more efficient and reliable contemporary software development process is presented. The results obtained as a result of the advance are detailed. It also discusses how lived experiences can be found in other industries and in the theoretical frameworks previously presented.

Post-implementation, I conducted an in-depth evaluation of performance metrics to gauge the success of integrating Agile, DevOps, and Cloud Computing in company's development process. We evaluated key performance indicators (KPIs) such as release frequency, number of bugs encountered, and development speed. The notable reduction in bug frequency (from 72 to an average of four) and the refined release cycles (from one release needed for bug fixes every week down to one release every five months) reflect a direct correlation between the employed methodologies and enhanced software quality.

This digital and paradigm transformation was only possible with the adoption of various cloud services such as web applications, SQL databases, storage services, analytics and insights. In addition to these, an extremely important functionality that has started to be used on this occasion is the web application slots. Through it new functionalities can be tested by the development team in a work environment similar to the production one, but without affecting the users. Once the company assessed these changes, it smoothly integrated them into the main application, minimizing potential deployment hiccups. Additionally, the company standardized the offering of georeplication to reinforce data availability and disaster recovery, aligning with recommendations (Warner, 2020, p. 203).

The whole method highlighted the significant character of an organizational culture that recognizes change. It also features the necessity for a stability between technology, processes and people. Azure DevOps and Visual Studio served as the technological enablers, while the refined development and deployment processes mirrored DevOps principles. The efficiency and quality of delivered software have been greatly improved with the help of the human factor. Viewed in parallel, the experiences inside this company and those experienced by other organizations in the field, we distinguish the possibility of replicating such a transformation. The results obtained by Axes Software are part of those found in the specialized literature presented in the first section, which mention the possibility of future process optimizations. The transformation carried out so far is an important starting point for the company's future goals. These include advances in automation, application data collection and processing, and continued evolution to implement more industry standards and recommendations.

The knowledge gained as a result of this process provides a clear and pragmatic vision in relation to the integration of Azure, DevOps and Cloud Computing. Similar companies considering moving towards a modern and efficient model can use this knowledge as a guide.

According to Srithar et al. (2022, pg. 01-05), a significant transformation of a company such as the integration of new methodologies and technologies also has financial implications. They go beyond initial investments and move towards long-term economic results. Axes Software encountered both upfront and long-term costs that affected the company's evolution.

Axes Software made significant financial commitments during the initial phase of its transformation. These commitments included direct costs such as acquiring cloud services and necessary tools for the transition. However, the financial investment went beyond these tangible expenses to include subtler areas, such as employee training to adeptly handle the new technologies. An illustrative instance of this investment was the integration of Azure DevOps into

the company's operations. This commitment was evident in various forms, particularly through staff upskilling, which represented both a financial and strategic investment.

Axes Software invested approximately 8,000 euros in training programs and certifications, such as AZ-104 (Microsoft Azure Administrator), AZ-204 (Developing Solutions for Microsoft Azure), DP-300 (Administering Relational Databases on Microsoft Azure), and AZ-305 (Designing Microsoft Azure Infrastructure Solutions), to enhance the team's proficiency in the latest technologies and align their skills with industry best practices. The increase in the quality of the services offered and the operational efficiency were achieved through the efficient use of human and technical resources made available through Cloud Computing. The costs incurred were considered by the company to be a long-term investment. Among these are the consumption of hardware resources, the cost of licensing, support services, regular updates, security features necessary to maintain operational resiliency and relevance. Moreover, the partnership with Microsoft, amounting to 5,000 euros annually, provides all the licenses required. Additionally, the company allocates approximately 900 euros each month for cloud computing and an additional 149 euros for 4 parallel workers in Azure DevOps.

Through the implementation of new methodologies and technologies, financial efficiencies have been achieved that are worth noting. With a remarkable 275% increase in clients preferring the company's cloud-hosted solutions over their on-premises options, Axes has experienced substantial growth in its customer base. Considering the average income per contract of 450 euros and the addition of 11 new customers, the monthly benefit from these new contracts can be calculated. With a profit margin of 25%, each contract yields a profit of 112.50 euros. Therefore, the total monthly profit from these new contracts is 1,237.50 euros. At the same time, the implementation process was streamlined. Reduced the time required to install new versions from 11 days to just two. This streamlining has produced significant savings for the company. Given that Axes typically performs 12 upgrades each year for their customers and that a day of work costs 400 euros, the total annual savings can be calculated. Previously, the cost of deployment for one upgrade was 4,400 euros. With the new streamlined process, the cost per upgrade is reduced to 800 euros. Therefore, the total annual savings from deployment is 31,200 euros.

With an initial investment totalling approximately 25,588 euros, encompassing expenses for training programs, certifications, partnership with Microsoft, cloud computing, and additional computing power in Azure DevOps, the company has realized significant returns. Long-term success will be achieved through strategy based on continuous workforce improvement and adoption of industry best practices and new technologies. Furthermore, the increase in clients preferring the company's cloud-hosted solutions by 275% has contributed to substantial growth in the customer base, resulting in a monthly profit of 1,237.50 euros from new contracts. In addition, streamlining the implementation process produced other savings. They amount to a total annual value of 31,200 euros. As a result, the total annual financial benefit derived from these initiatives amounts to 46,050 euros. With a return on investment (ROI) of approximately 180.14% for the first year, Axes Software's strategic investment in Agile, DevOps, and Cloud Computing has proven to be highly lucrative and impactful.

The growing appetite of customers for hosting services in the cloud is mainly due to satisfaction and trust. It reflects the safety and reliability of the team of software applications and cloud infrastructure management services provided by Axes Software. In addition, investment in training and certification programs for staff has yielded benefits in the form of improved skills. As a result, an increase in operational efficiency and a reduction in the costs of development and

attracting new customers was noted. Drawing the line, we notice both financial gains, customer satisfaction, increased safety, and reliability, as well as a more qualified and efficient technical team. All this contributes to the continuous success of Axes Software in the Auto Id market in Romania.

Conclusions

The integration of Agile, DevOps and Cloud Computing requires the transition from simple development, with a high risk of error and high costs to a new digital era in which the processes of creating applications become more and more efficient and with multiple long-term advantages. Axes Software represents a convincing case study for the implementation of a method that combines the organization, the human and the technical, leading to performance:

1. Integration as a stimulus for change: Axes Software's experience with Agile methodologies, Cloud Computing and DevOps techniques demonstrates the positive and significant impact they have on software development. This impact represents a profound transformation in the way programmers work.
2. Strategic and cultural change: Axes Software shows how not only the implementation of new technologies and work methodologies, but also cultural integration and adaptation to diversity lead to continuous success.
3. Empirical benefits and financial viability: Despite the high level of investment, Axes Software shows that through this integration, in the long term, the quality of services and profit increase substantially, and the number of errors decreases significantly, making the time allocated and the effort made more efficient.
4. Continuous learning and adaptation: As technological evolution happens at an unprecedented speed, the example of Axes Software demonstrates the importance of continuous adaptation and research, the thirst for knowledge of an entire team.
5. A model for others: the experiences and the path taken ranks Axes Software as a potentiator for all companies in order to transition from an ineffective way of working to another that ensures the quality of learning, the quality of work and visibility on the competitive market.
6. Looking ahead: Over time, continuing in the same integrated model, Axes Software keeps in everyone's attention the pursuit of digital excellence and the orientation towards continuous expansion and maintains the practice of perfecting these working techniques.

The integration of Agile, DevOps, and Cloud Computing transcends simple trendiness in software development; it represents a strategic necessity in the digital age. Axes Software's expedition offers a compelling case study of this integration's tangible implementation and advantages. It underscores the significance of adopting a comprehensive approach that encompasses technology, processes, and people, furnishing invaluable insights for any organization striving to excel in the continuously evolving realm of technology and software development.

References

Amazon. (2022). *On-Demand Infrastructure on AWS Helps Capital One DevOps Teams Move Faster Than Ever*. (Amazon) Retrieved 10 01, 2023, from https://aws.amazon.com/solutions/case-studies/capital-one-devops/

Azad, N. (2022). Understanding DevOps critical success factors and organizational practices. *2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB)*, *1*, pp. 83-90. Pittsburgh, PA, USA. doi:10.1145/3524614/3528627

Bucena, I., & Kirikova, M. (2017). Simplifying the DevOps Adoptation Process. *BIR Workshops*, (pp. 1-15). Aberdeen, UK.

Chang, C.-Y., Ou, P.-P., & Deng, D.-J. (2019). Cross-Site Large-Scale Software Delivery with Enhanced Git Branch Model. *IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, (pp. 153-156). Beijing, China. doi:10.1109/ICSESS47205.2019.9040834

Clivati, P. (2021, 02 15). *How Etsy CEO Structures Teams to Optimize for Experimentation and Agility*. (Growth Hackers) Retrieved 10 01, 2023, from https://blog.growthhackers.com/how-etsy-ceo-structure-teams-to-optimize-for-experimentation-and-agility-e3455348bddf

Diel, E., Marczak, S., & Cruzes, D. (2016). Communication Challenges and Strategies in Distributed DevOps. *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, (pp. 24-28). Orange County, CA, USA. doi:10.1109/ICGSE.2016.28

Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software, 123*(1), 176-189.

Hyrynsalmi, S. M. (2022). The State-of-the-Art of the Integration Platforms as a Service research. *2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB)*, (pp. 17-22). Pittsburgh, PA, USA. doi:10.1145/3524614.3528634

Khan, A., Keung, J., Niazi, M., Hussain, S., & Zhang, H. (2017). Systematic Literature Reviews of Software Process Improvement: A Tertiary Study. *Software and Services Process Improvement: 24th European Conference, EuroSPI 2017*. Ostrava, Czech Republic. doi:10.1007/978-3-319-64218-5_14

Khan, M., Khan, A., Khan, F., Khan, M., & Whangbo, T. (2022). Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic Review. *IEEE Access, 10*(1), 14339-14349. doi:10.1109/ACCESS.2022.3145970

Klotins, E., & Peretz-Andersson, E. (2022). The unified perspective of digital transformation and continuous software engineering. *2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB)*, *1*, pp. 75-82. Pittsburgh, PA, USA. doi:10.1145/3524614.3528626

Marnewick, C., & Langerman, J. (2020). DevOps and organisational performance: the fallacy of chasing maturity. *IEEE Software, 38*(5), 48-55.

Nitu, A. (2022). Ensuring Data Security in Cloud Computing. *IBIMA Computer Science Conference.* Granada, Spain.

Sravan, S. S., Ganesh, C. S., Kiran, K., Chandra, T. A., Aparna, K., & Vignesh, T. (2023). Significant Challenges to espouse DevOps Culture in Software Organisations By AWS: A methodical Review. *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, (pp. 395-401). Coimbatore, India. doi:10.1109/ICACCS57279.2023.10113021

Srithar, S., Vetrimani, E., Vignesh, V., Ulaganathan, M., Kumar, B., & Alagumuthukrishnan, S. (2022). Cost-Effective Integration and Deployment of Enterprise Application Using Azure Cloud Devops. *2022 International Conference on Computer Communication and Informatics (ICCCI)*, (pp. 01-05). Coimbatore, India. doi:10.1109/ICCCI54379.2022.9740874

Srithar, S., Vetrimani, E., Vignesh, V., Ulaganathan, M., Rejesh Kumar, B., & Alagumuthukrishnan, S. (2022). Cost-Effective Integration and Deployment of Enterprise Application Using

Azure Cloud Devops. *2022 International Conference on Computer Communication and Informatics (ICCCI)*. Coimbatore, India. doi:10.1109/ICCCI54379.2022.9740874

Stoica, M., Ghilic-Micu, B., & Mircea, M. (2013). Software Development: Agile vs. Traditional. *Informatica Economică*(December). doi:10.12948

Sutherland, J. (2014). *Scrum, the Art of Doing Twice the Work in Half the TIme* (2014 ed.). New York: Crown Business.

Tang, S., Lee, B.-S., & He, B. (2014). Towards Economic Fairness for Big Data Processing in Pay-as-You-Go Cloud Computing. *IEEE 6th International Conference on Cloud Computing Technology and Science*, (pp. 638-643). Singapore. doi:10.1109/CloudCom.2014.120

Wang, H., Jing, Q., Chen, R., He, B., Qian, Z., & Zhou, L. (2010). Distributed systems meet economics: Pricing in the cloud. *2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*. Boston, MA, USA. Retrieved from https://www.usenix.org/legacy/event/hotcloud10/tech/full_papers/WangH.pdf

Warner, T. L. (2020). *Microsoft Azure For Dummies* (2nd ed., Vol. 1). 1119612144: For Dummies.