How DevOps Impacts Enterprise Architecture In the Banking and Financial Services Industry

Jiaxin Cui
University of Melbourne
cindycui.cn@gmail.com

Linh Luong
University of Melbourne
lihmono@gmail.com

Minh Huong Nguyen
University of Melbourne
vicky.ngminh@gmail.com

Nadya Aurelia Herryyanto
University of Melbourne
nadyaaureliah@gmail.com

Ngan Nga Pham
University of Melbourne
ngangan.pham@gmail.com

Rod Dilnutt
University of Melbourne
rpd@unimelb.edu.au

Abstract

DevOps, an agile approach that emphasises collaboration, automation, and continuous improvement in software development, has garnered significant traction in the banking and financial services industry. Despite its benefits, there remains a lack of comprehensive research on the interaction between DevOps and Enterprise Architecture (EA). To address this gap, this paper utilises systematic literature review to identify the core practices of DevOps and analyse the implications of DevOps on EA, including agility, scalability, resilience, resource utilisation, and collaboration. The research then explores the alignment of DevOps impacts with established EA Frameworks like Zachman Framework®[1](ZF) and The Open Group Architecture Framework (TOGAF®[2]). Through a comprehensive case analysis of Capital One, the paper further illustrates how DevOps implementation can influence EA in the banking sector. The outcome provides valuable insights into the benefits of DevOps adoption within EA paradigms for banking enterprises while noting potential challenges to implementation.

1. Introduction

In a Harvard Business Review survey, 86% of business professionals believed that rapid software development and deployment was important for their company's success; however, only 10% suggested their organisations were doing so, creating a competitive advantage gap (Competitive Advantage through DevOps, 2019). DevOps, an approach that facilitates collaboration between the development and operation of applications, emerges as a solution. This practice helps organisations deploy new products quicker than using conventional software development processes (AWS, 2019).

The banking and financial services industry, characterised by decades-old operations, grapples with complex legacy systems requiring modernisation. In this industry, a common practice for EA is the monolithic approach, which centralises all core software and functionality into an integrated application (Aydemir & Basciftci, 2022). Over time, the complex and incomprehensible codebase becomes challenging to implement additional features, hindering new technology adoption (Aydemir & Basciftci, 2022). To solve the monolithic limitations, some banks have adopted agile methodology, microservices and DevOps to increase their flexibility and innovation (Bucchiarone et al., 2017).

While DevOps presents an opportunity to enhance product development, there is a research gap regarding its limited adoption in the banking and financial services industry. As legacy technology is among the major challenges in transforming EA (Competitive Advantage through DevOps, 2019), further research on DevOps and its impacts on EA

[1] The Zachman Framework is a registered trademark of Zachman International.
[2] TOGAF is a registered trademark of The Open Group

is crucial for enhancing IT infrastructure and process efficiency. Likewise, the lack of empirical evidence regarding the impact of DevOps on EA in the banking domain highlights a gap that this paper seeks to address. Thus, this paper aims to answer the following question: “How does DevOps impact enterprise architecture?” to help bridge the gap between TOGAF® and ZF frameworks and modern DevOps methodologies, facilitating smoother integration of DevOps in the banking and financial services industry.

To answer the research question, this proposal will first conduct a systematic literature review on the core practices of DevOps and its impacts on EA. It will later explore the congruence between identified DevOps impacts with ZF and TOGAF® and illustrate literature findings in a case analysis. The paper will finally discuss implications, limitations and directions for future research on this topic.

2. Research Methodology

Following Okoli and Schabram’s suggestion (2015), this paper employs the Systematic Literature Review (SRL) to explore two main research scopes:

1. What is DevOps, and what is its intersection with EA?
2. Using TOGAF® and ZF, a compatibility evaluation of DevOps’s impacts on banking’s EA will be conducted.

Relevant literature was discovered through two main sources: Google Scholar; and the University of Melbourne Library, and screened for quality based on different criteria. The majority of articles were selected based on publication within the last fifteen years in high-quality journals and websites in the fields of information systems, information technology, or financial services. Then, the article content had to be related to DevOps and other EA concepts. Additionally, data collection methodology and sample size should also be considered. After screening, 63 articles were analysed, synthesised, and discussed in the sections below.

3. Literature Review

3.1. DevOps

DevOps addresses the siloed structure of traditional software delivery, where Development and Operations teams work independently, leading to miscommunication and goal misalignment (Qumer Gill et al., 2018; Azad & Hyrynsalmi, 2023). DevOps addresses these issues by bridging the connection and enhancing the efficiency and quality of software delivery (Trigo et al., 2022).

3.1.1. Principles

DevOps principles are the fundamental ideas that form the DevOps approach.

- **Collaboration:** DevOps emphasises shared responsibilities and team collaboration within organisations (Maroukian & Gulliver, 2020; Luz et al., 2018).

- **Automation:** DevOps values the ability to eliminate monotonous tasks and maximise automation during the Software Development Lifecycle (SDLC) (Qumer Gill et al., 2018).

- **Continuous Enhancement:** DevOps focuses on continuous improvement to continually adapt to changing circumstances (Jha et al., 2023).

- **Customer-Oriented Actions:** DevOps facilitates an early feedback loop with end users through rapid deployment and real-time monitoring (Qumer Gill et al., 2018).

- **Create With The End In Mind:** DevOps requires teams to have a holistic understanding of the output they are producing, from creation to deployment (Jha et al., 2023).

3.1.2. DevOps Pipeline

A DevOps pipeline is a set of automated processes that facilitates the collaboration between the development and operations teams (Jha et al., 2023; Trigo et al., 2022).

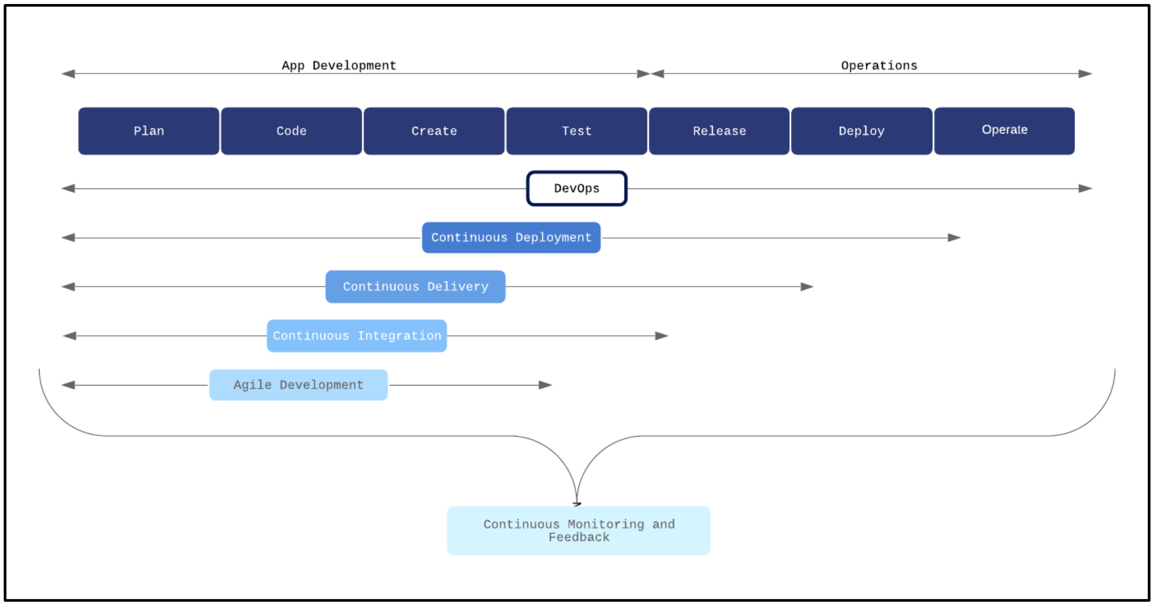


Figure 1. *DevOps Process Flow*

Note. From *Understanding The DevOps Process Flow* by Lucidchart. https://www.lucidchart.com/blog/devops-process-flow. Copyright 2024 Lucid Software Inc.

DevOps pipeline typically includes the following components:

- **Continuous Integration (CI):** The practice of automatically merging code changes into a shared repository (Trigo et al., 2022).

- **Continuous Delivery (CD):** The practice of testing and ensuring the readiness of the compiled code to be released through manual approval (Trigo et al., 2022).

- **Continuous Deployment (CD):** The practice of automatically validating and deploying the new version of software without human intervention (Trigo et al., 2022).

- **Continuous Monitoring and Feedback:** The practice of automatically assessing the health and performance of the software (Jha et al., 2023).

3.1.3. Enablers

The increasing interest in DevOps aligns with the shift towards microservices architecture for enterprise agility (Pianini & Neri, 2021; Waseem et al., 2020). Unlike monolithic architecture, which integrates all software components into a single unit, microservices architecture divides software into smaller, independent services, promoting flexibility and scalability (Mooghala, 2023; Pianini & Neri, 2021).

Similarly, cloud computing and containerisation contribute significantly to DevOps adoption (Jha et al., 2023; Senepathi et al., 2019). Cloud computing, which is the ability to use computing services through the Internet, provides DevOps with continuously available automation capabilities (Jha et al., 2023). Containerisation, the practice of encapsulating software with all the necessary dependencies needed for its execution, enables faster deployment since software packages can be deployed without further configuration (Jha et al., 2023).

3.2. Impacts of DevOps on EA

The integration of DevOps practices in EA reshapes conventional paradigms of software development and deployment. Our systematic literature review reveals the following impacts of DevOps on EA.

3.2.1. Agility and Speed

Traditionally, software development involves time-consuming activities like manual testing and integration, causing deployment delays (Mohammad, 2018; Qumer Gill et al., 2018). DevOps addresses these challenges by streamlining processes and accelerating SDLC via automation, eliminating manual interference (Gregory & Crispin, 2014; Laan, 2011; Nybom et al., 2006). As such, DevOps expedites time-to-market, enabling agile responses to market demands, enabling continuous innovation, and efficient customer value delivery (Colavita, 2016; Farroha & Farroha, 2014; Faustino et al., 2022; Fitzgerald & Stol, 2017; Mohammad, 2018; Perera et al., 2017; Wiedemann et al., 2023). This shift impacts EA by necessitating a flexible and modular architecture that supports frequent, incremental updates and rapid changes.

3.2.2. Scalability and Flexibility

Traditional monolithic applications often face challenges like costly scaling and performance optimisation (Premchand & Choudhry, 2018). Scalability and flexibility are fundamental impacts of DevOps on EA, addressing the need for systems to handle increasing workloads, adapt to changing requirements, and accommodate growth in terms of users, data, and infrastructure (Jaju, 2023). DevOps practices such as CI/CD pipelines, microservices adoption, containerisation, and orchestration enable scalability and flexibility by reducing lead times, increasing stability and reliability, and facilitating on-demand scaling and cost control (Forsgren et al., 2018; Jaju, 2023; Pahl et al., 2017; Di Francesco, 2017). In EA, this requires the shift from monolithic architectures to decentralised, service-based architectures where components can be scaled independently based on demand.

3.2.3. Improved Reliability and Resilience

DevOps significantly improves reliability and resilience in EA, overcoming challenges like component replacement difficulties and complex application maintenance in monolithic systems (Premchand & Choudhry, 2018). DevOps environments, particularly those incorporating CI and microservices, enable organisations to track, measure, and increase reliability across software programs in isolation, reducing downtime and failure risks associated with deployments (Oviedo, 2021; Luz et al., 2019; Di Francesco, 2017). Additionally, resilience is achieved through auto-scaling and recovery automation, enabling applications to adapt quickly to adverse situations and automatically recover from failures (Luz et al., 2019; Di Francesco, 2017). For EA, this translates to a need for resilient designs that can withstand failures and recover quickly.

3.2.4. Resource Utilisation

DevOps significantly impacts resource utilisation in EA, leveraging technological optimisations like containerisation, orchestration tools, and cloud computing services (Premchand & Choudhry, 2018). DevOps and microservices architecture enable each service to use its technology stack, empowering teams to select appropriate technologies for each service and improving resource allocation and adaptability (Di Francesco, 2017). Moreover, microservices promote efficient resource utilisation by enabling the reuse of existing services for developing new functionality or applications (Di Francesco, 2017). This requires EA to be built on an infrastructure that supports elastic resource allocation, leveraging cloud-native architectures to optimise cost and performance.

3.2.5. Collaboration

DevOps fosters improved collaboration in EA by breaking down silos, promoting cross-functional teams, and integrating operations tasks into development activities (Masombuka & MnKandla, 2018; Walls, 2014; Othman et al., 2016; Chan & Thong, 2007). This collaborative culture reduces delays and conflicts, ensuring a smoother software delivery process (Luz et al., 2019). Through shared responsibility and continuous improvement, DevOps enhances efficiency and effectiveness in EA (Hermawan & Manik, 2021; Farroha & Farroha, 2014; Perera et al., 2017). As such, EA needs to support cross-functional workflows by promoting transparency and shared services across departments.

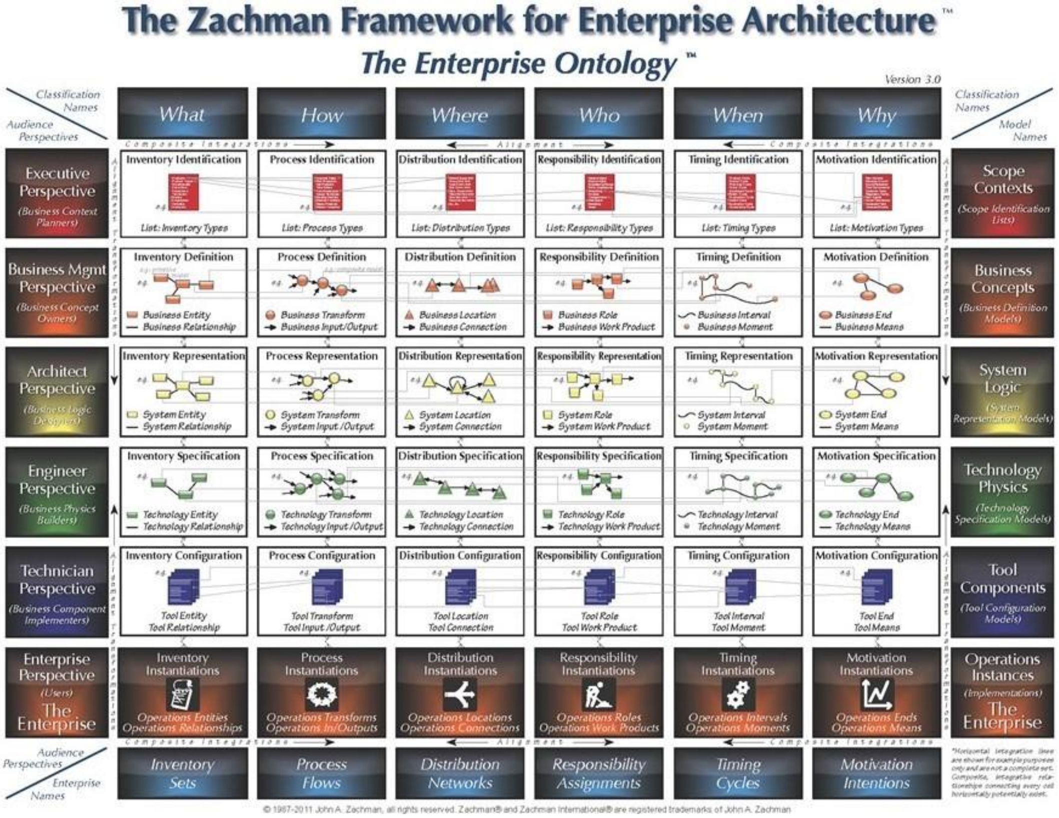
Hence, DevOps fundamentally shifts the requirements of enterprise architecture by driving the need for agility, scalability, reliability, resource efficiency, and collaboration. As organisations adopt DevOps, EA must evolve to support these principles through more modular, flexible, and responsive designs.

3.3. Analysing the Congruence between DevOps Impact and EA Frameworks: Zachman Framework and TOGAF® (10th Edition)

This section explores the impacts of DevOps on two prominent EA frameworks and its compatibility with established structures. By examining how DevOps aligns with these frameworks, organisations can ensure that their existing architectures effectively accommodate DevOps principles. The choice to focus on ZF and TOGAF® stems from their widespread acceptance and application within the Banking and Financial Services industry (Gokhale, 2010; Kotusev, 2021; Sun & Chen, 2010).

3.3.1. Zachman Framework (ZF)

ZF is a classification structure containing descriptive representations of an enterprise's complex IT systems and processes (Iyamu, 2018; Anthony Jnr et al., 2021). As illustrated in Figure 2, the structure of ZF is divided into six rows representing different perspectives relative to EA and six columns symbolising different aspects of EA (Jafari et al., 2009; Zachman, 2008).


The Zachman Framework for Enterprise Architecture ™
The Enterprise Ontology ™
Version 3.0
Composite Integrations
Alignment
Alignment
Transformations
*Horizontal integration lines are shown for example purposes only and are not a complete set. Composite, integrative relationships connecting every cell horizontally potentially exist.
© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

Classification Names Audience Perspectives	What	How	Where	Who	When	Why	Classification Names Model Names
Executive Perspective (Business Context Planners)	Inventory Identification List: Inventory Types	Process Identification List: Process Types	Distribution Identification List: Distribution Types	Responsibility Identification List: Responsibility Types	Timing Identification List: Timing Types	Motivation Identification List: Motivation Types	Scope Contexts (Scope Identification Lists)
Business Mgmt Perspective (Business Concept Owners)	Inventory Definition Business Entity Business Relationship	Process Definition Business Transform Business Input/Output	Distribution Definition Business Location Business Connection	Responsibility Definition Business Role Business Work Product	Timing Definition Business Interval Business Moment	Motivation Definition Business End Business Means	Business Concepts (Business Definition Models)
Architect Perspective (Business Logic Designers)	Inventory Representation System Entity System Relationship	Process Representation System Transform System Input /Output	Distribution Representation System Location System Connection	Responsibility Representation System Role System Work Product	Timing Representation System Interval System Moment	Motivation Representation System End System Means	System Logic (System Representation Models)
Engineer Perspective (Business Physics Builders)	Inventory Specification Technology Entity Technology Relationship	Process Specification Technology Transform Technology Input /Output	Distribution Specification Technology Location Technology Connection	Responsibility Specification Technology Role Technology Work Product	Timing Specification Technology Interval Technology Moment	Motivation Specification Technology End Technology Means	Technology Physics (Technology Specification Models)
Technician Perspective (Business Component Implementers)	Inventory Configuration Tool Entity Tool Relationship	Process Configuration Tool Transform Tool Input /Output	Distribution Configuration Tool Location Tool Connection	Responsibility Configuration Tool Role Tool Work Product	Timing Configuration Tool Interval Tool Moment	Motivation Configuration Tool End Tool Means	Tool Components (Tool Configuration Models)
Enterprise Perspective (Users) The Enterprise	Inventory Instantiations Operations Entities Operations Relationships	Process Instantiations Operations Transforms Operations In/Outputs	Distribution Instantiations Operations Locations Operations Connections	Responsibility Instantiations Operations Roles Operations Work Products	Timing Instantiations Operations Intervals Operations Moments	Motivation Instantiations Operations Ends Operations Means	Operations Instances (Implementations) The Enterprise
Audience Perspectives Enterprise Names	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	

Figure 2. *Zachman Framework*

Note. From *About The Zachman Framework* by John A. Zachman. https://zachman-feac.com/zachman/about-the-zachman-framework. Copyright 2008 Zachman International®, Inc.

3.3.1.1. Agility, Speed and Innovation

DevOps positively impacts the agility of the ZF. Specifically, practices such as CI/CD enhance data processing and analysis (Jaju, 2023). In Function, DevOps streamlines and automates development, testing, and deployment processes (Nybom et al., 2006), delivering functionality quicker. In People, DevOps, with its automation capabilities, increases developers's problem-solving and quality of work (Laan, 2011). The practice also facilitates collaboration between technical and operational teams (Skekic et al., 2018), enhancing communication speed and avoiding silos. Together, DevOps enhances product development to meet business goals, fulfilling the Motivation aspect in the ZF. DevOps furthermore enhances innovation in the ZF as it encourages experimentation and continuous learning (Jha et al., 2023), therefore facilitating innovation within the workforce. The efficient communication between technical and operational teams, facilitated by rapid feedback cycles (Qumer Gill et al., 2018), also enables better ideation and implementation of innovative ideas and market changes.

3.3.1.2. Scalability and Flexibility

Scalability, the ability to accommodate growth in users, data and infrastructure, is a critical aspect of DevOps (Bahadori & Vardanega, 2019). Microservices architecture and containerisation allow scalable data structure (What?) and application/functions (How?), while cloud computing offers a dynamic data storage and network infrastructure to manage traffic fluctuation, regardless of location (Where?). DevOps also promotes shared responsibilities and cross-functional knowledge (Who?), enhancing team flexibility and adaptation to changes (Vu, n.d.). With automation, continuous enhancement and rapid deployment, DevOps practices allow banks to achieve faster time-to-market (When?) and increase competitiveness through innovation and productivity (Babbar et al., 2023).

3.3.1.3. Improved Reliability and Resilience

The DevOps approach offers a reliable system that encourages integrating security solutions at every stage of the CI/CD pipeline, enhancing customer protection (David et al., 2024). Automated data backups and disaster recovery processes also ensure data availability and minimise system recovery cycles. Moreover, CI/CD practices promote system testing to detect and address vulnerabilities and failure points, reducing functional errors (How?) and improving system reliability. Thus, DevOps secures customer transactions and privacy, uninterrupted access, and minimal downtime, creating reliability for the system.

3.3.1.4. Resource Utilisation

DevOps utilises human (Who?) and time (When?) resources by automating tasks in a development cycle, thereby reducing manual tasks and idle time while boosting productivity. Moreover, cloud-based infrastructure eliminates the need for physical hardware management, further saving costs for physical resources. From the motivation and strategy aspect (Why?), DevOps' scalability and flexibility allow the banking industry to optimise resource allocation and cost management, leading to better performance and profitability (Joe, 2023).

3.3.1.5. Collaboration

DevOps facilitates collaboration in the ZF as it fosters cross-functional collaboration between developers and operations teams (Skekic et al., 2018) and enhances communication among geographically dispersed teams through cloud-based platforms. Most importantly, DevOps increases collaboration in the ZF's People aspect and connects different perspectives for a common goal of rapid software development and deployment. Real-time communication and feedback loops within DevOps teams also streamline decision-making and problem-solving (Qumer Gill et al., 2018), therefore benefitting the Time aspect.

3.3.2. TOGAF®

TOGAF® is an EA development framework that supports diverse styles of architecture (Buchanan, 2010; The Open Group®, 2024). The Architecture Development Model (ADM) forms the core of TOGAF®, providing an iterative, step-by-step approach to EA development and management (The Open Group®, 2024). Other key components include

Enterprise Continuum, which provides methods to categorise architecture and solution artefacts and Architecture Repository, which offers a structured framework to store and manage architectural assets (The Open Group®, 2024).

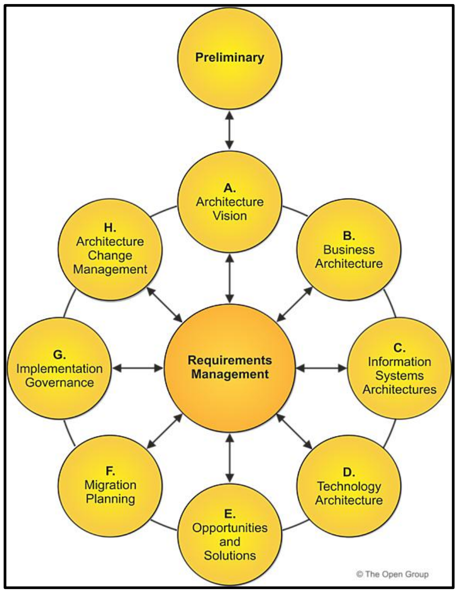

Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management
© The Open Group

Figure 3. *Architecture Development Model*
Note. From *Architecture Development Method (ADM)* by The Open Group®. https://pubs.opengroup.org/togaf-standard/adm/chap01.html. Copyright 2022 The Open Group®.

3.3.2.1. Agility, Speed and Innovation

DevOps's impact on agility and speed aligns with TOGAF® as its framework encompasses methods for optimising processes and expediting development cycles (The Open Group®, 2024). For instance, TOGAF's® ADM underscores the significance of iterative and incremental development, mirroring DevOps' emphasis on CD. Moreover, automation, a core tenet of DevOps, is also emphasised in TOGAF®. Specific guidance on iterative and incremental development, aligning with DevOps principles, is provided in TOGAF's® "Applying the TOGAF® ADM using Agile Sprints," detailed in Part 3.2.

3.3.2.2. Scalability and Flexibility

DevOps' impact on scalability and flexibility aligns with TOGAF's® guidance on designing scalable and flexible architectures, including considerations like modularity and interoperability (The Open Group®, 2024). TOGAF® emphasises standardisation and interoperability, facilitating the integration of DevOps tools and technologies for smooth scalability. These principles are detailed in TOGAF's® Part III: ADM Guidelines and Techniques, particularly in Sections 2 and 7, focusing on architecture principles and flexible integration.

3.3.2.3. Improved Reliability and Resilience

DevOps' enhanced reliability and resilience align with TOGAF's® methodologies for resilient architecture design, minimising downtime and ensuring system stability (The Open Group®, 2024). TOGAF's® focus on risk management and fault tolerance aligns with DevOps' objectives of mitigating failures and maintaining system stability. This is reflected in Part IV: Architecture Content and EA Context, which covers architecture governance and risk management. This synergy empowers organisations to effectively address potential points of failure (Mueller et al., 2013).

3.3.2.4. Resource Utilisation

The impact of DevOps on resource utilisation directly affects TOGAF's® ADM, particularly in phases related to technology selection (Phase B), architecture design (Phase C), and implementation (Phase D). This alignment with

TOGAF's® resource management frameworks enables organisations to optimise resource utilisation, minimise waste, and achieve cost efficiencies throughout their IT infrastructure (The Open Group®, 2024). These principles are detailed in Part V: Enterprise Continuum and Tools of TOGAF®, which delves into selecting technologies and tools for resource optimisation.

3.3.2.5. Collaboration

DevOps' impact on improved collaboration aligns with various aspects of TOGAF®. For instance, in Phase A: Architecture Vision of TOGAF's® ADM, identifying stakeholders and their concerns is emphasised (The Open Group®, 2024). While primarily intended for architecture development, these guidelines can be extrapolated to aid in stakeholder management during the implementation of DevOps within an enterprise (Nakakwa et al., 2013). Similarly, TOGAF's® Part VII: Architecture Capability Framework covers skills and competencies needed for collaborative architectures. However, TOGAF® only addresses this impact on a meta-level, as it is not directly addressed within TOGAF®. Nevertheless, it can be mitigated by the general features of frameworks, such as promoting a common mindset or language (Mueller et al., 2013).

3.4. Comparative Analysis of TOGAF® and ZF

This table illustrates a comparison between the two discussed EA frameworks, evaluating their alignment with the impacts of DevOps.

Impacts	ZF	TOGAF
Agility and Speed	Yes	Yes
Scalability and Flexibility	Yes	Yes
Improved Reliability and Resilience	Yes	Yes
Resource Utilisation	Partial	Yes
Collaboration	Partial	Partial

Table 1. *Evaluation of the Alignment Between EA Frameworks and DevOps Impacts*

TOGAF® is more suited to the continuous, iterative nature of DevOps because it is designed to adapt and evolve, while Zachman’s more rigid, structured approach may struggle to accommodate the constant changes and feedback cycles central to DevOps practices.

4. Case Analysis – Capital One’s DevOps Adoption

This section offers a case analysis of Capital One, a leading US bank, using information from the official websites of Amazon Web Services (AWS) and Capital One to validate literature review findings in practical scenarios.

4.1. How Capital One Employs DevOps

Capital One strategically adopted DevOps to enhance competitiveness and respond to evolving customer demands while maintaining its speed and quality in software delivery. With a prioritisation in response to customer feedback, the bank transitioned from on-premises data centres to a cloud-first strategy with AWS, aiming to foster innovation and rapid iteration. Transitioning to microservices architecture and integrating agile practices and automation, Capital One streamlined its development and deployment from 2012 to 2020, demonstrating a strong commitment to technological innovation and customer satisfaction.

4.2. How DevOps impacted on Capital One

4.2.1. Agility and Speed

Capital One dramatically increased agility, software development and deployment speed by adopting DevOps and automating business processes. The disaster recovery time in tests was reduced by 70%, and development time was reduced from three months to only minutes. The company also accelerated its innovation pace, going from quarterly and monthly application updates to more frequent releases. This was achieved by empowering teams to take ownership of their software delivery processes, allowing rapid response to market changes and customer feedback.

4.2.2. Scalability and Flexibility

Migrating to the public cloud and adopting microservices architecture enabled Capital One to scale resources dynamically based on demand. This flexibility allowed it to handle user traffic more efficiently, ensuring high availability of services and adapting to changing market needs correspondingly. For instance, by using AWS's cloud storage, Capital One meets fluctuating demand reliably, benefiting from 99.99% data durability. It enables easy data management at any scale with robust access controls, flexible replication tools, and organisation-wide visibility.

4.2.3. Improved Reliability and Resilience

As part of its transition from monolithic to microservices architecture, Capital One implemented automated CI/CD pipelines. By enforcing rigorous compliance gates, including source control mechanisms, secure storage, access control, quality and security checks, the bank ensured that every deployment adhered to predetermined standards, reducing failure risks. It is complemented with pipeline templates, enabling faster and more reliable deployments by eliminating manual processes and bottlenecks. Moreover, the use of reusable building blocks and flexible orchestration options ensured resilience and reliability throughout the software delivery lifecycle.

4.2.4. Resource Utilisation

DevOps optimised resource utilisation at Capital One by automating repetitive tasks, streamlining processes, and leveraging cloud infrastructure efficiently. The organisation completed the migration from eight on-premises data centres to AWS, fully embracing the cloud to optimise resource utilisation, ensuring that computing and storage resources are efficiently allocated to meet business needs. Additionally, as mentioned above, reusable building blocks and flexible orchestration options maximised productivity and minimised operational costs.

4.2.5. Collaboration

DevOps fosters a culture of collaboration and cross-functional teamwork at Capital One. It removed silos and created team ownership within the organisation. It also expanded the company’s use of open-source products and its participation in open-source communities to gain faster access to ideas and talent.

4.3. Alignment of Capital One's DevOps Adoption with TOGAF® & ZF

This table illustrates Capital One's path to DevOps and assesses how its approach aligns with the ZF and TOGAF®.

Impact	Capital One	ZF	TOGAF
Agility, Speed and Innovation	Emphasis on iterative and incremental development processes and automation to accelerate development cycles via CI/CD pipelines.	Partial	Yes
Scalability and Flexibility	Emphasis on modular architectures and interoperability via microservices and cloud adoption.	Yes	Yes
Improved Reliability and Resilience	Enforced compliance gates and a certification system for pipelines and deployment strategy.	Partial	Partial

Resource Utilisation	Migrated its data centre to the cloud, paying only for the computing and storage its applications require.	Yes	Yes
Collaboration	Removed silos and created team ownership.	Partial	Partial

Table 2. *Evaluation of the Alignment Between EA Frameworks and DevOps Impacts in Capital One*

Based on the above analysis, Capital One's DevOps adoption strategies align more closely with TOGAF® than the Zachman Framework (ZF). Capital One's alignment with TOGAF® suggests that its enterprise architecture is designed to handle the fast-paced, iterative nature of modern software development, ensuring smoother integration of DevOps practices.

5. Discussion

5.1. Implication

This paper enhances the understanding of DevOps and its implication for EA, particularly in the banking sector. It entails how DevOps practices can address the challenges of monolithic architectures and how DevOps drives digital transformation in financial institutions using the ZF and TOGAF®. Moreover, this paper suggests that the impacts of DevOps align more closely with TOGAF® rather than ZF. As such, TOGAF® is generally regarded as more adaptable for integrating DevOps into enterprise architecture compared to ZF.

However, organisations should carefully evaluate their specific requirements before selecting the most appropriate framework. TOGAF's® adaptability supports DevOps by facilitating agility and rapid response to change, while Zachman may require extensions or hybridisation to integrate DevOps methodologies effectively. In some cases, combining TOGAF's® operational flexibility with Zachman's strategic oversight could provide a balanced approach, allowing organisations to leverage the strengths of both frameworks. Ultimately, framework selection or modification depends on the organisation's goals and its need for agility in software development processes.

Additionally, by employing SLR, the research demonstrates the effectiveness of interdisciplinary approaches in analysing the intersection of DevOps and EA in banking, identifying gaps and offering methodological insights for future research.

5.2. Limitation and Recommendations

Despite the significant benefits, this paper acknowledges some limitations that should be addressed or researched further.

1. **DevOps adoption:** This paper has yet to consider confounding factors affecting DevOps adoption, such as resistance to change, traditional culture, and lack of understanding of DevOps. Thus, future research should adopt a holistic approach that considers both technical and non-technical factors impacting the adoption of DevOps.

2. **Methodological constraints:** The research scope's banking focus may hinder the discovery of relevant studies and the findings' generalisability across industries. Future research can conduct cross-sector comparative analyses for broader insights into DevOps' impact on EA in diverse organisational contexts.

3. **Insufficient case analysis depth:** Although this paper analyses a specific case, CapitalOne's transformation mainly focuses on cloud adoption through DevOps. Consequently, the scope of real-world effects, successes, and challenges of DevOps in banking remains narrowly defined and hard to gauge. Future research should delve into detailed banking sector case studies to enhance understanding of effective DevOps implementation.

6. Conclusion

This research explores the impact of DevOps on EA in the banking and financial services sector. Through a systematic literature review and case analysis of Capital One's DevOps adoption, the paper reveals DevOps' significant potential to enhance agility, scalability, reliability, resource utilisation, and collaboration within EA frameworks. The comparative analysis of DevOps impacts on the ZF and TOGAF® underscores greater alignment between DevOps practices and the methodologies advocated by TOGAF®. Capital One's case analysis further reinforces this alignment, demonstrating how the organisation effectively leveraged DevOps to foster innovation and prioritise customer-centricity within its EA, and highlights the broader resonance of this organisation's approach with TOGAF® principles. Overall, the research contributes to a deeper understanding of DevOps and its implications for EA, providing valuable insights for academia and industry alike. By embracing DevOps principles and methodologies, financial institutions can unlock new opportunities for efficiency, innovation, and long-term success in a rapidly evolving digital landscape.

7. Bibliography

Amazon Web Services. (2019). *What is DevOps?* https://aws.amazon.com/devops/what-is-devops/

Amazon Web Services. (2020). *Capital One Completes Migration from Data Centers to AWS, Becomes First US Bank to Announce Going All In on the Cloud.* https://aws.amazon.com/solutions/case-studies/capital-one-all-in-on-aws/

Amazon Web Services. (2020). *Capital One's Tech Transformation Enables Improved Agility and Innovation.* https://aws.amazon.com/solutions/case-studies/capital-one-tech-transformation-enables-improved-agility-innovation/?did=cr_card&trk=cr_card

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

Amazon Web Services. (2024). *"How to Cloud" with Capital One.* https://aws.amazon.com/solutions/case-studies/capital-one-enterprise/

Amazon Web Services. (2024). *Amazon S3 Features.* https://aws.amazon.com/s3/features/?pg=ln&sec=be

Anthony Jnr, B., Abbas Petersen, S., Helfert, M. and Guo, H. (2021). Digital transformation with enterprise architecture for smarter cities: a qualitative research approach. *Digital Policy, Regulation and Governance, 23*(4), 355-376. https://doi.org/10.1108/DPRG-04-2020-0044

Aydemir, F. & Basciftci, F. (2022). Building a Performance Efficient Core Banking System Based on the Microservices Architecture. *Journal of Grid Computing*, *20*(4), 1-37, doi:10.1007/s10723-022-09624-z.

Azad, N., & Hyrynsalmi, S. (2023). DevOps critical success factors — A systematic literature review. *Information and Software Technology*, *157*. https://doi.org/10.1016/j.infsof.2023.107150

Babbar, A., Janardhanan, R., Paternoster, R., & Soller, H. (2023, April 11). Why most digital banking transformations fail and how to flip the odds. *McKinsey & Company*. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-forward/why-most-digital-banking-transformations-fail-and-how-to-flip-the-odds

Bahadori, K., & Vardanega, T. (2019). DevOps meets dynamic orchestration. *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, *11350*, 142–154. https://doi.org/10.1007/978-3-030-06019-0_11

Bente, S., Bombosch, U. & Langade, S. (2012). 'Chapter 2 - What Is Enterprise Architecture?', in Bente, S., Bombosch, U., & Langade, S. (eds) *Collaborative Enterprise Architecture*, Morgan Kaufmann, Boston, doi:10.1016/B978-0-12-415934-1.00002-9.

Bucchiarone, A., Dragoni, N., Dustdar, S., Larsen, S.T. & Mazzara, M., (2017). *From Monolithic to Microservices: An experience report*. Technical University of Denmark, Örebro University, TU Wien, Danske Bank, and Innopolis University.

Buchanan, S. (2010). Planning Strategically, Designing Architecturally: A Framework for Digital Library Services, Woodsworth, A. (Ed.) Advances in Librarianship (Advances in Librarianship, Vol. 32), Emerald Group Publishing Limited, Leeds, 159-180. https://doi.org/10.1108/S0065-2830(2010)0000032010

Capital One. (2019). *Powering Capital One's Microservices Journey with CICD Pipelines.* https://www.capitalone.com/tech/software-engineering/realigning-devops-practices-to-support-microservices/

Capital One. (2020). *Capital One Completes Migration from Data Centers to AWS, Becomes First US Bank to Announce Going All In on the Cloud*. https://aws.amazon.com/solutions/case-studies/capital-one-all-in-on-aws/

Chan, K. Y., & Thong, J. Y. (2007). An integrated framework of individual acceptance of agile methodologies. *PACIS 2007 Proceedings*, 154. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1130&context=pacis2007

Colavita, F. (2016). DevOps Movement of Enterprise Agile Breakdown Silos, Create Collaboration, Increase Quality, and Application Speed. In: Ciancarini, P., Sillitti, A., Succi, G., Messina, A. (eds) Proceedings of 4th International Conference in Software Engineering for Defence Applications. Advances in Intelligent Systems and Computing, vol 422. Springer, Cham. https://doi.org/10.1007/978-3-319-27896-4_17

Competitive Advantages through DevOps (2019). Harvard Business Review Analytics Services. https://inthecloud.withgoogle.com/hbr-report-19/hbr-research-report-google-cloud.pdf

David, P., Kushwaha, M. K., & Suseela, G. (2024, March 22). *DevSecOps in finance: Strengthening the security model of applications*. 4th International Conference on Data Engineering and Communication Systems, Bangalore, India. https://doi.org/10.1109/icdecs59733.2023.10502917

Di Francesco, P. (2017, April). Architecting microservices. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) (pp. 224-229). IEEE. https://ieeexplore.ieee.org/iel7/7957715/7958419/07958492.pdf?casa_token=0Pdd19r8RnYAAAAA:e_Wv_7lNYntmETyVYYm6NjH9d-K_o6m1x7FStka7fVqCYOo4HN6u9FezKNK4JmwCCOgWf7KX

Draheim, D & Weber, G eds. (2007). Trends in Enterprise Application Architecture: 2nd International Conference. *TEAA 2006.* Berlin, Germany.

Farroha, B. S., & Farroha, D. L. (2014, October). A framework for managing mission needs, compliance, and trust in the DevOps environment. In *2014 IEEE Military Communications Conference* (pp. 288-293). IEEE. https://ieeexplore.ieee.org/iel7/6954705/6956719/06956773.pdf?casa_token=Q58r7HEnwTQAAAAA:OBw1slIVsYJf3TMYDIGor5pMvtOFiBYhQQZanoet3yUwRJARDwqfgaded7omkGLxXruMZXJO

Faustino, J., Adriano, D., Amaro, R., Pereira, R., & da Silva, M. M. (2022). DevOps benefits: A systematic literature review. *Software: Practice and Experience*, *52*(9), 1905-1926. https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3096?casa_token=FvU8zytJm9cAAAAA:ENSIHSo9dKavbBYFCjwAwK_u5dzh_RALaw2vH2BH4ZngShocTF1o_fzal2fqfHXYPPd7UYEinne0tQ

Fitzgerald, B., & Stol, K. J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, *123*, 176-189. https://www.sciencedirect.com/science/article/pii/S0164121215001430?casa_token=bVys8S6buowAAAAA:HFo896AgMrsTvK-5LxCInHiS24AwFxf7clufslqK7XrgpiqE0TmhpCIZb24Fe7ls9VHpsyMnwQ

Forsgren, N., Smith, D., Humble, J., & Frazelle, J. (2019). 2019 Accelerate State of DevOps report. *DORA and Google Cloud, Tech. Rep*, *48455*. http://research.google/pubs/2019-accelerate-state-of-devops-report/

Gerber, A, le Roux, P, Kearney, C & van der Merwe, A (2020). *The Zachman Framework for Enterprise Architecture: An Explanatory IS Theory*. University of Pretoria, South Africa.

Gokhale, A. (2010). Increasing effectiveness of the Zachman framework using the balanced scorecard. Purdue University. https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1028&context=techmasters

Gregory, J., & Crispin, L. (2014). *More agile testing: learning journeys for the whole team*. Addison-Wesley Professional. https://books.google.com.au/books?hl=en&lr=&id=uq-pBAAAQBAJ&oi=fnd&pg=PR9&dq=Gregory,+J.,+and+Crispin,+L.,+2015,+More+Agile+Testing:+Learning+Journeys+for+the+Whole+Team,+Addison-Wesley,+Upper+Saddle+River,+N.J.&ots=9DegXMPhCP&sig=MdFZZE79A-Quwemv1oxMi0cm2LA

Hermawan, A., & Manik, L. P. (2021). The effect of DevOps implementation on teamwork quality in software development. *Journal of Information Systems Engineering and Business Intelligence*, *7*(1), 84. https://e-journal.unair.ac.id/JISEBI/article/download/25904/13996

Iyamu, T. (2018). Implementation of the enterprise architecture through the Zachman Framework. *Journal of Systems and Information Technology*, *20*(1), 2-18. https://doi.org/10.1108/JSIT-06-2017-0047

Jafari, M., Akhavan, P. and Nouranipour, E. (2009). Developing an architecture model for enterprise knowledge: An empirical study based on the Zachman framework in Iran. *Management Decision, 47*(5), 730-759. https://doi.org/10.1108/00251740910960097

Jaju, I. (2023). Maximizing DevOps Scalability in Complex Software Systems: Maximizing DevOps Scalability in Complex Software Systems. https://www.diva-portal.org/smash/get/diva2:1787653/FULLTEXT01.pdf

Jha, A. V., Teri, R., Verma, S., Tarafder, S., Bhowmik, W., Kumar Mishra, S., Appasani, B., Srinivasulu, A., & Philibert, N. (2023). From theory to practice: Understanding DevOps culture and mindset. *Cogent Engineering, 10*(1). https://doi.org/10.1080/23311916.2023.2251758

Joe, T. (2023). Cloud computing and information systems: Enabling scalability and flexibility. *Business Studies Journal*, *15*(3), 1–3. https://www.abacademies.org/articles/cloud-computing-and-information-systems-enabling-scalability-and-flexibility-16121.html#r2

Kotusev, S. (2021, July 1). A comparison of the top four enterprise architecture frameworks. *BCS*.https://www.bcs.org/articles-opinion-and-research/a-comparison-of-the-top-four-enterprise-architecture-frameworks/#:~:text=Presently%2C%20TOGAF%20has%20achieved%20the

Laan, S. (2017). *IT Infrastructure Architecture-Infrastructure Building Blocks and Concepts Third Edition*. Lulu. com. https://books.google.com.au/books?hl=en&lr=&id=LDf6DQAAQBAJ&oi=fnd&pg=PR15&dq=Laan,+S.2011,+%E2%80%9CIT+infrastructure+architecture:+Infrastructure+building+blocks+and+concepts,%E2%80%9D+U.S.A:+Lulu+Press.&ots=sjLHjiGF7I&sig=_PXhYQXaYR7UGaw-tFVDP4u0jic

Lucidchart. (2024). *Understanding the DevOps process flow* (2024). https://www.lucidchart.com/blog/devops-process-flow

Luz, W. P., Pinto, G., & Bonifácio, R. (2018). Building a Collaborative Culture: A Grounded Theory of Well Succeeded DevOps Adoption in Practice. In *ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (ESEM '18)*. Oulu, Finland.

Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, *157*, 110384. https://www.sciencedirect.com/science/article/pii/S0164121219301517?casa_token=2gB9XSLj_FMAAAAA:8HltRv-MJFMGZdshvcDXSvk5cxIJ2Nv4yXeSi8VJepedXhs7GGKbdR0WQjpE9L1y5Gj-jgg8znpu

Maroukian, K. and Gulliver, S. (2020) Leading Devops practice and principle adoption. In: 9th International Conference on Information Technology Convergence and Services (ITCSE '20)., 30-31 May 2020, Vancouver, Canada. http://centaur.reading.ac.uk/92771/

Masombuka, T., & Mnkandla, E. (2018, September). A DevOps collaboration culture acceptance model. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*. https://dl.acm.org/doi/pdf/10.1145/3278681.3278714?casa_token=X-SUdm-qmkIAAAAA:ozUhk5Pv1gogc5FNvfl0yw1gbyviooviLhxblWVjqko0v31lHCmqi4SAKAFQKl9O4XiJrG4ACbjW

Mohammad, S. M. (2018). Improve Software Quality through practicing DevOps Automation. *International Journal of Creative Research Thoughts (IJCRT)*. https://www.researchgate.net/profile/Sikender-Mohsienuddin-Mohammad/publication/343922189_Improve_Software_Quality_through_practicing_DevOps_Automation/links/5f484daa458515a88b795a37/Improve-Software-Quality-through-practicing-DevOps-Automation.pdf

Mooghala, Sridhar. (2023). A Comprehensive Study of the Transition from Monolithic to Micro services-Based Software Architectures. *Journal of Technology and Systems, 5*, 27-40. 10.47941/jts.1538.

Mueller, T., Schuldt, D., Sewald, B., Morisse, M., & Petrikina, J. (2013). Towards inter-organizational enterprise architecture management-applicability of TOGAF 9.1 for network organizations. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=82117169fda164493a41e18754a30e010e0b11eb

Nakakawa, A., Bommel, P. V., & PROPER, H. E. (2013). Supplementing enterprise architecture approaches with support for executing collaborative tasks—A case of TOGAF ADM. *International Journal of Cooperative Information Systems*, *22*(02), 1350007. https://www.worldscientific.com/doi/abs/10.1142/S021884301350007X

Nybom, K., Smeds, J., & Porres, I. (2016). On the impact of mixing responsibilities between devs and ops. In *Agile Processes, in Software Engineering, and Extreme Programming: 17th International Conference, XP 2016, Edinburgh, UK*. Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-319-33515-5_11

Okoli, C., Schabram, K. (2010). A Guide to Conducting a Systematic Literature Review of Information Systems Research. *Sprouts: Working Papers on Information Systems, 10*(26). http://sprouts.aisnet.org/10-26

Othman, H. B., Zouaoui, M., & Hamdoun, M. (2016, April). Organizational culture and the acceptance of agile methodology. In 2016 International Conference on Digital Economy (ICDEc) (pp. 16-23). IEEE. https://ieeexplore.ieee.org/abstract/document/7563140/

Oviedo, E. I. (2021, May). Software Reliability in a DevOps Continuous Integration Environment. In *2021 Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-4). IEEE. https://ieeexplore.ieee.org/iel7/9605695/9605657/09605768.pdf?casa_token=QwqUOJuoZBAAAAAA:2HLxFhhTbLr7A2OnV14rRtDzaAXfSjt-y_mqwkDhnFDzfJWBM7ISyPPaq8u9-MLhKJ6NIu-JDDg

Pahl, C., Brogi, A., Soldani, J., & Jamshidi, P. (2017). Cloud container technologies: a state-of-the-art review. *IEEE Transactions on Cloud Computing*, *7*(3), 677-692. https://ieeexplore.ieee.org/abstract/document/7922500/?casa_token=dUsDNkKNkRsAAAAA:XAawoLyS3mR5OMEvH-diIN8j9k2wG8yNALVQyVfc-sK0t2C7VIAyh7Pz6tRHb4Q_nodsoYMy

Perera, P., Silva, R., & Perera, I. (2017, September). Improve software quality through practicing DevOps. In *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 1-6).

Pianini, D., & Neri, A. (2021). Breaking down monoliths with Microservices and DevOps: an industrial experience report. 2021 *IEEE International Conference on Software Maintenance and Evolution (ICSME).* Luxembourg. doi: 10.1109/ICSME52107.2021.00051.

Premchand, A., & Choudhry, A. (2018, February). Architecture simplification at large institutions using microservices. In 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT). *IEEE*, 30-35. https://ieeexplore.ieee.org/document/8668173

Qumer Gill, A., Loumish, A., Riyat, I., & Han, S. (2018). DevOps for information management systems. *VINE Journal of Information and Knowledge Management Systems*, *48*(1), 122–139. https://doi.org/10.1108/vjikms-02-2017-0007

R, T (2019). *The Relationship Between Enterprise Architecture and DevOps*, https://www.leanix.net/en/blog/the-relationship-between-enterprise-architecture-and-devops

Scekic, M., Gazivoda, M., Scepanovic, S., & Nikolic, J. (2018). Application of DevOps approach in developing business intelligence system in bank', in *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, doi:10.1109/MECO.2018.8406047.

Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018.

Sun, J., & Chen, Y. (2010, August). Building a common enterprise technical architecture for an universal bank. In *2010 International Conference on Management and Service Science* (pp. 1-4). IEEE. https://ieeexplore.ieee.org/iel5/5575324/5575325/05577437.pdf?casa_token=REGmT9Y0dj4AAAAA:n-XnLdIrfypvf2ogFwXqsgUj5h47uAxJtJS0V72ytPxgZW10ak_V7140XWP7zxxzO5Qk6UdP

The Open Group. (2022). Architecture Development Method (ADM). The Open Group. https://pubs.opengroup.org/togaf-standard/adm/chap01.html

The Open Group. (2024, May). The TOGAF® Standard — Digital Edition - Introduction. The Open Group. https://pubs.opengroup.org/togaf-standard/

Trigo, A., Varajão, J., & Leandro, S. (2022). DevOps adoption: Insights from a large European Telco. *Cogent Engineering*, *9*. https://doi.org/10.1080/23311916.2022.2083474

Vu, S. (n.d.). *The importance of team structure in DevOps*. Atlassian. https://www.atlassian.com/devops/frameworks/team-structure

Walls, M. (2013). *Building a DevOps culture*. O'Reilly Media.

Waseem, M., Liang, P., & Shahin, M. (2020). A Systematic Mapping Study on Microservices Architecture in DevOps. *Journal of Systems and Software, 170*. https://doi.org/10.1016/j.jss.2020.110798

Wiedemann, A., Wiesche, M., Gewald, H., & Krcmar, H. (2023). Integrating development and operations teams: A control approach for DevOps. *Information and Organization*, *33*(3), 100474. https://www.sciencedirect.com/science/article/pii/S1471772723000283?casa_token=3XZnFkFLEM8AAAAA:kiswS31znIE_fCOmRZRJFnCklcc3Z8pDLgJR-8e6vvU0AaW3qsbceTkI_epevts7Eoz3PNVdYjyn

Zachman, J. A. (2008). About The Zachman Framework. Zachman International Enterprise Architecture. https://zachman-feac.com/zachman/about-the-zachman-framework