2020

Strategies for IT Product Managers to Manage Microservice Systems in Enterprises

Walter Zrzavy
Walden University

Walden University

College of Management and Technology

This is to certify that the doctoral study by

Walter Zrzavy

has been found to be complete and satisfactory in all respects,
and that any and all revisions required by
the review committee have been made.

Review Committee
Dr. Jodine Burchell, Committee Chairperson, Information Technology Faculty
Dr. Charlie Shao, Committee Member, Information Technology Faculty
Dr. Steven Case, University Reviewer, Information Technology Faculty

Chief Academic Officer and Provost
Sue Subocz, Ph.D.

Walden University
2020

Abstract

Strategies for IT Product Managers to Manage Microservice Systems in Enterprises

by

Walter Zrzavy

MSc, Donau Universität Krems, 2007

Doctoral Study Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Information Technology

Walden University

June 2020

Abstract

Many information technology (IT) product managers have experienced significant challenges in adopting microservice architecture (MSA) systems successfully in their organizations. Inefficiencies resulting from MSA system adoption are of concern to IT product managers as these inefficiencies increase the cost of maintenance and increase the time to deliver software updates to the business. Grounded in the technology-organization-environment theory, the purpose of this qualitative multiple case study was to explore the strategies used by IT product managers for the domain-driven design and development and operations practices to reduce the inefficiencies during the MSA system adoption. The participants were 18 IT product managers associated with the adoption and operation of microservice architecture systems at a global consumer goods manufacturer company and a global financial company, both with headquarters in Europe. Data were collected from semi structured interviews and a review of 7 documents. A thematic analysis was used to analyze the data. Four major themes emerged to include organizational alignment in adopting MSA, ways of working, experienced-based approach to design MSA systems, and MSA environment landscape. A key recommendation for IT product managers is to adopt an IT organization structure aligned with the business context of the MSA system allowing for a full lifecycle approach. The implications for positive social change include the potential for IT product managers to improve the work environment for the MSA-related teams, which may lead to robust software systems and easier to use applications by removing barriers and increasing accessibility for users, thus supporting individuals in their daily life.

Strategies for IT Product Managers to Manage Microservice Systems in Enterprises

by

Walter Zrzavy

MSc, Donau Universität Krems, 2007

Doctoral Study Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Information Technology

Walden University

June 2020

Dedication

I dedicate this work to my wife Christina and my son Julian for their continuous support and understanding of the long hours of absence from the family life while I was working on this study.

Acknowledgments

First, I want to thank my wife, Christina, for supporting my pursuit of this degree. Without her encouragement to continue during difficult times and her sacrifice due to my absence from the family while working on this study, it would have been impossible to complete this endeavor. This study would have unimaginable without her.

Secondly, I would like to thank the gatekeepers for enabling access as a research site and advertising my research. Without their support, this study would not have provided insights into the challenges and strategies applied for MSA-based systems. Also, my sincere appreciation to all participants who made time available and shared their experience in support of my research. Without their help, my study would not have been successful.

Last but not least, I would like to thank Dr. Jodine Burchell, my committee chair, Dr. Charlie Shao, my second committee member, as well as Dr. Steven Case, my University Research Reviewer, for their time to review and providing detailed feedback to improve my study. Their guidance and encouragement allowed me to complete my doctoral study.

Table of Contents

List of Tables

List of Figures

Section 1: Foundation of the Study

Background of the Problem

The microservices architecture (MSA) style emerged from the field of software practitioners to solve customer problems. This architecture style was a new approach to service-oriented architecture (SOA) that implements an application as a set of distributed small services (Pautasso, Zimmermann, Amundsen, Lewis, & Josuttis, 2017). The MSA style is promising due to less complicated software development cycles, flexible integration of new functionalities, performance scalability, and easier maintenance (Alshuqayran, Ali, & Evans, 2016; Lewis & Fowler, 2016). However, the downside of using the MSA style is that it introduces a set of new challenges for enterprises in the development, deployment, and operations of microservices (O'Connor, Elger, & Clarke, 2017a). The focus on the agile development of business capabilities, using component-based services realized as microservices with an autonomous character that is continuously evolving, requires new ways of data management, consistency of data models, and recovery approaches (Bogner & Zimmermann, 2016; Ebert, Gallardo, Hernantes, & Serrano, 2016). Due to these challenges, organizations may struggle to identify strategies to adopt microservices into their IT landscape and operate these services efficiently to gain the expected benefits. While research studies address system quality, cloud, and migration problems (Francesco, Lago, & Malavolta, 2019), there is currently limited understanding in the research community of the best practices in MSA-based software systems adoption. Understanding these challenges and exploring what strategies IT product managers apply in practice will fill this research gap.

Problem Statement

The increased usage of microservice architecture (MSA) systems has resulted in challenges for IT managers to employ best practices for the successful adoption of MSA in an organization (Zimmermann, 2016). Between 2016 to 2017, practitioners reported bad practices with 71% reporting wrong cuts as harmful related to the MSA design approach, and 57% report shared persistency as harmful for development and operations (DevOps) of resulting in the inefficient adoption of MSA (Taibi & Lenarduzzi, 2018). The general IT problem is that inefficiencies resulting from MSA system adoption increase the cost of maintenance and increase the time to deliver software updates to the business. The specific IT problem is that some IT product managers lack strategies for the domain-driven design and DevOps practices to reduce the inefficiencies in the MSA system adoption.

Purpose Statement

The purpose of this qualitative multiple case study was to explore strategies for the domain-driven design and DevOps practices used by IT product managers to reduce the inefficiencies in the MSA system adoption. The population for this study was IT product managers and DevOps members associated with the adoption and operation of microservice architecture systems at a global consumer goods manufacturer company and a global financial company within Europe. The potential social impact of this study includes the possible improvement of the user experience for software systems constructed as MSA used by individuals in their daily life.

Nature of the Study

I chose the qualitative methodology to explore strategies for the domain-driven design and DevOps practices used by IT product managers to reduce the inefficiencies in the MSA systems adoption. Qualitative research is needed when a researcher is exploring a phenomenon where the complexity of the problem calls for an in-depth investigation (Hazzan & Nutov, 2014). Qualitative methodology was appropriate because of my intention to gain a detailed, in-depth understanding of strategies used by IT managers to adopt microservice architecture systems in their organizations that enabled benefits for their organization. Quantitative research is suitable for studies involving variables, where the relationship of attributes can be measured, analyzed, and evaluated by using statistical analysis (McCusker & Gunaydin, 2015). Quantitative methodology was not suitable for this study, as I did not want to investigate the relationship between variables such as IT managers' strategies and inefficiencies. A mixed-method, a combination of qualitative and quantitative methods, was also not the best approach for this study. For example, the mixed method may be used when the research question requires a qualitative exploration to provide insights and subsequently to confirm the model with quantitative methods (Molina-Azorin, 2016). The mixed-method was not suitable for my study as I did not create a theory to be tested by a quantitative method. Thus, because my study calls for an in-depth exploration rather than the use of variables and statistical analysis, the qualitative method was best suited to investigate the domain-driven design and DevOps practices of IT managers employing for the successful MSA adoption.

I chose a multiple case study design for my research study. A case study design is suitable when the researcher is investigating contemporary events in its natural environment within a bounded context and has limited or no control over the related behavior (Yin, 2014). The multiple case study design was appropriate for this study because the investigation did provide insights into similarities and differences between the cases of different industries for the domain-driven design and DevOps strategies that IT managers apply in their environment for the MSA adoption.

Other potential designs I could have chosen included ethnographic, phenomenological, and narrative designs. The ethnographic design is suitable for studies that try to understand a culture or cultural group (Grossoehme, 2014). The ethnographic design was not appropriate, as my research was not concerned about the culture of the IT product managers. The phenomenological design can be used to explore the understanding of a shared lived experience of participants related to a phenomenon (Skea, 2016). I understood that each IT manager experienced the phenomenon under study differently; therefore, the phenomenological study was not appropriate for my research. Narrative design is concerned with the understanding of the individual's life, describing the actions, events, and situations in stories (Gergen, Josselson, & Freeman, 2015). The narrative design was not suitable as I did not describe the life of the IT managers but the strategies they use. The most appropriate qualitative research design for my study was a multiple case study design, as I wanted to gain a detailed understanding of the common strategies used by IT managers for the adoption of a complex phenomenon in organizations.

Research Question

What strategies do IT product managers used for the domain-driven design and DevOps practices to reduce the inefficiencies in the MSA system adoption?

Interview Questions

1. What inefficiencies or technical debt do you and your team experience using the domain-driven design approach in the development of MSA-based applications?
2. What domain-driven-design strategies do you use to identify, reduce, or prevent inefficiencies or technical debt of MSA-based applications?
3. Which domain-driven-design strategies are most effective in reducing the inefficiencies of MSA-based applications?
4. What challenges do you and your team experience using DevOps for MSA-based applications?
5. What DevOps practices do you and your team apply for MSA-based applications?
6. Which DevOps strategies were most effective in reducing the inefficiencies of MSA-based applications?
7. What, if any, other inefficiencies or challenges do you experience in the adoption of MSA systems?
8. What strategies do you apply for adopting MSA-based system in your organization?

9. What factors do you apply in selecting the domain-driven design strategies and DevOps practices for the MSA system adoption?

Conceptual Framework

I used the technology-organizational-environment (TOE) theory to conduct my research and evaluate the data collected. Tornatzky and Fleischer developed the framework in 1990 to explain how the context of a company influences the adoption of innovation. The framework defines (a) the technological context that includes relevant technologies and processes within and outside the organization, (b) the organizational context related to characteristics and resources of the enterprise, and (c) the environmental context of the industry that impacts the decision-making process for introducing the innovation to the organization (Tornatzky & Fleischer, 1990). The TOE model is used in information technology research to explain the adoption of a broad range of information systems within a host of industries and allows the researcher to vary the factors for different innovations (Awa, Ojiabo, & Orokor, 2017a).

The TOE framework allowed me to investigate the domain-driven design and DevOps strategies of IT managers for the adoption of MSA using the three components of technical factors, organizational factors, and environmental factors. Figure 1 presents the TOE model and factors for the adoption of technological innovation. Bradford, Earp, and Grabski (2014) investigated the factors to be considered for the implementation of a centralized end-to-end identity and access management and Enterprise Resource Planning (ERP) systems using TOE in a multi-case study design. Similarly, (Kurnia, Karnali, & Rahim, 2015) employed a multi-case study design with TOE theory as a leading research framework to guide the investigation of influential business-to-business electronic commerce (EC) technologies adoption factors. Based on suitability in previous research, I determined the TOE framework was appropriate as the domain-driven design is a technology concept to design microservices, and DevOps incorporates organizational measures for the adoption of MSA.

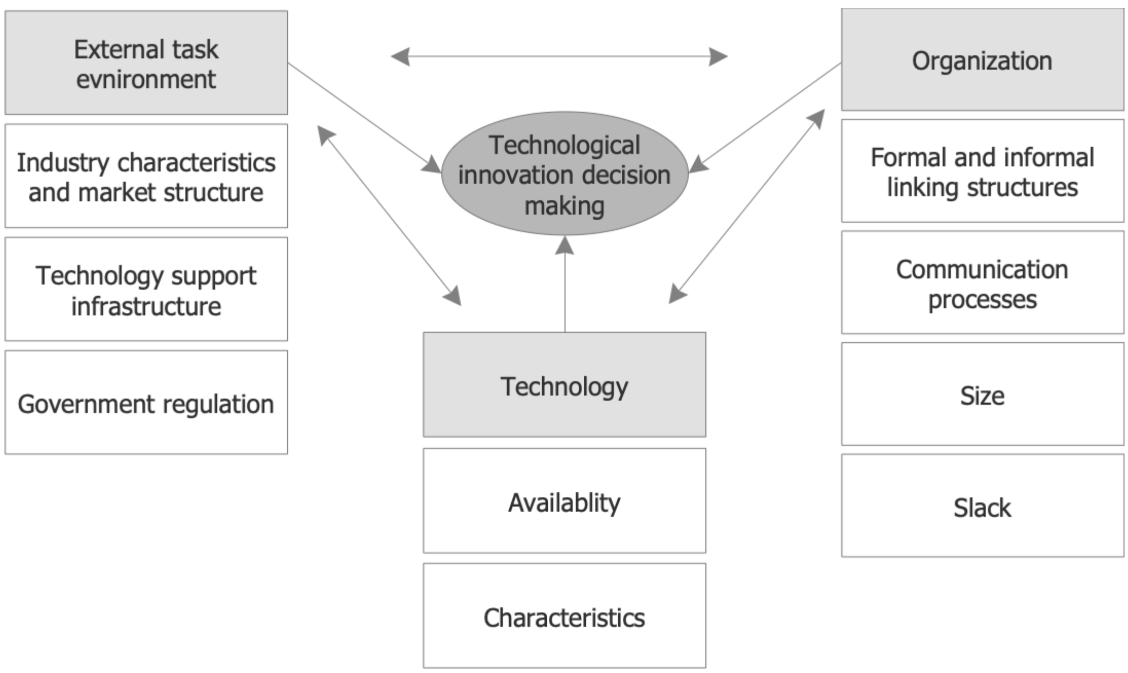

Figure 1. The technology-organization-environment (TOE) framework and context factors. Adapted from Tornatzky and Fleischer (1990, p. 153).

Definition of Terms

The following industry-specific terms are used and defined where no clear definition is established.

Container: A container is a self-contained unit that is part of the application architecture deployment structure and realized to operate most efficiently, contained, and portable (Kratzke & Quint, 2017). Containers are typical deployment units for microservice services. The realization is based on virtualization on operating-system-level and does not emulate full hardware to ensure less usage of resources (Amaral et al., 2015; Hassan & Bahsoon, 2016).

Continuous Delivery (CD): Continuous delivery enables on-demand deployment of software in an automated way and is part of DevOps practice (Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016).

Continuous Integration (CI): Continuous integration supports the merging of code from many developers frequently during a day and to perform code validations (Shahin, Ali Babar, & Zhu, 2017).

Development and Operations (DevOps): DevOps is an approach to streamline the integration of the software development process with the implementation and operations of the same (Kang, Le, & Tao, 2016).

Domain-driven design: Domain-driven design captures domain concepts, such as business practices and languages, in software design models described with classes, attributes, and methods (Rademacher, Sorgalla, & Sachweh, 2018). Domain-driven design is a model-based-development approach to create bounded organization context for software development and software integration (Jamshidi, Pahl, Mendonca, Lewis, & Tilkov, 2018).

Infrastructure as a Code (IaC): Infrastructure as code is an approach to automate the infrastructure deployment and configuration in support of the lifecycle of the software (Jiang & Adams, 2015). IaC is the support environment for DevOps, CD, and microservice architecture style service.

*Microservices: "*Microservice architectural style is an approach to developing a single application as a suite of small services, each running in its process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies and use different data storage technologies." (Lewis & Fowler, 2016).

Assumptions, Limitations, and Delimitations

Assumptions

Assumptions are unproven conditions to a study that are treated to be true without verification (Foss & Hallberg, 2014). The following assumptions are taken for this study. I assumed that the two case organizations were representative of the industry. I assumed that the participants that I selected represented the respective case and have an end-to-end experience in domain-driven design and DevOps practices in adopting MSA. I assumed that the participants had sufficient experience in the context microservices to provide quality information for the phases in architecting, development, deployment, operations, and maintenance. I assumed that the number of interviews and other sources of evidence provided appropriate information for domain-driven design approaches and DevOps practices in the context for MSA to achieve quality in the research outcome.

Limitations

Limitations are restrictions in the study that are beyond of the researcher's control (Denscombe, 2013). The limitations of this study were influenced by the qualitative approach for this study. The interpretation of information into themes derived from interviews and other sources is a subjective process due to potential bias that influences validity and reliability (Yin, 2014). Additionally, a limitation for the study was the cases of two industries that may not allow transferring the findings to other industries or cases. Another limitation was the selection of the IT managers involved in the domain-driven design and DevOps practices of adoption of the MSA system that was responsible for only for a subset of MSA systems.

Delimitations

Delimitations are boundaries that are constraints for the study defined by the researcher to enable the completion of the study (Denscombe, 2013). I set the following delimitations for the study. The cases of the study were confined to organizations based in Europe to minimize my travel expenditure. The selection of organization was based on my level of access to the leadership in the information technology group for a global consumer goods manufacturer company and a financial company. The interview participants were in the role of IT manager for MSA systems only and were in this position for at least 1 year. The case companies had at least 15 MSA systems in productive operation for at least 2 years.

Significance of the Study

Contribution to Information Technology Practice

The increased adoption of MSA systems is redefining the way business organizations are interacting with IT organizations. The decision to adopt MSA systems requires having strategies in place to realize the benefits for the organization (Killalea, 2016). This study contributed to the IT practice because it provided detailed descriptions of strategies that are used to adopt MSA-based systems in organizations. The strategies included, but were not limited to, (a) approaches for decentralized data management of the business domain driven microservice designs, (b) approaches of handling the most critical data and business processes with the required security and compliance needs based on alignment to the business value chain, (c) strategies for testing to increase the change rate and decrease the time to repair, (d) strategies for the capabilities and setup of teams adopting MSA, and (e) strategies to develop the toolchain for the continuous integration and continuous deployments of MSA. Therefore, this study contributed to the existing literature and provided more knowledge to IT managers with a detailed description of successful domain-driven design and DevOps strategies for MSA adoption.

Implications for Social Change

The findings of this study may support positive social change as more useful MSA-based software applications can be provided to individuals that can be used in a wide range of activities from social networking, online shopping, streaming of videos, financial transactions, and other potential applications. The adoption of MSA-based systems may allow organizations to enhance and improve existing applications faster that, in turn, simplifies user activities in the individual's daily life.

A Review of the Professional and Academic Literature

The purpose of this qualitative multiple case study was to explore strategies used by IT product managers for the adoption of MSA systems in their enterprise IT. MSA was a recent trend of SOA with increasing adoption across industries to facility agile delivery methods with a highly flexible service orientation. My focus for the review of the literature was MSA, the context for the adoption of MSA development and operations, and the TOE framework.

My review of the academic and professional literature included sources from IEEE Xplore, ProQuest Central, Science Direct, ACM Digital Library, Google Scholar, and EBSCO Host as the primary search locations. As the research on my research topic emerged, I explored the reference list of articles to identify alternate sources. I identified the peer-review status of each paper by searching the paper's international standard serial number (ISSN) or journal-title in the Ulrich's Global Serials Directory.

My literature search included the review of seminal material and relevant peer-reviewed articles with a focus on recent articles from 2014 and newer. I used a combination of terms and phrases that evolved as a strategy for searching the literature, including *microservice*, *DevOps, agile, domain-driven*, *model-driven, bounded context, service-oriented architecture, SOA, MSA, technology organization environment,* and *TOE*. The literature review included 107 articles and three seminal sources, with 103

(96.26%) as peer-reviewed. A total of 89 (83.18%) of papers were published within the last 5 years of my anticipated study completion year of 2020.

The term and concept of *MSA* was introduced and adopted by practitioners in 2014 and thus the research on MSA increased between 2014 to 2018. However, the majority of papers were published as conference papers with a limited number of peer-reviewed articles. The most dominant research methodology used in research on MSA is the qualitative approach. Additionally, I performed a systematic literature review investigating the current state of the research on MSA and provided a view of classifications for these studies. My study focused on the strategies for domain-driven design and DevOps related to MSA adoption. I could not identify research with a focus on strategies to adopt MSA systems. What I identified was research that presented recommendations for specific problems of MSA, and migrations from a monolithic architecture to MSA-based systems. Therefore, I based the structure of the literature review on the most dominant topics investigated in the peer-reviewed papers and the concern of my research question.

The review of the literature has five major components (a) TOE, (b) MSA, (c) domain-driven design, (d) DevOps, and (e) container technology. The review of the TOE framework focused on the related theories, the limitations, the usage in the adoption research, and the service-oriented architecture research. Additionally, I presented how the TOE theory will support investigations into the strategies of domain-driven design and DevOps practices for the MSA-based system adoption. The research into MSA included the history, the characteristics, the SOA, technologies, positioning of MSA and SOA, context, and adoption strategies. The domain-driven-design component focused on the MSA-based system design aspects and related strategies. The DevOps component presented the current state of practices relevant to MSA-based systems. The container technology component focused on the usage relevant to design MSA-based systems and DevOps practices.

Technology-Organizational-Environment (TOE) Framework

The TOE framework was developed by Tornatzky and Fleischer in 1990 to describe the influence of an enterprise's environment on the adoption of technological innovation. Tornatzky and Fleischer (1990) identified three contextual elements and factors that influence the organizational adoption process: the technological context, the organizational context, and the external task environmental context. Also, Tornatzky and Fleischer (1990) stated that a broader context exists that influences the adoption, implementation, and usage for technological innovations as constraints or benefits. The TOE framework allows the researcher to focus on the technological, organizational, and environmental factors that are relevant for a stakeholder to develop strategies in the adoption of technology innovations. The TOE framework is the result of an investigation into the process of innovation and context of the organization.

The contextual structure of TOE provides a framework for researcher to investigate the adoption of technology in organizations. Tornatzky and Fleischer (1990) described the current state of research on innovations at various stages from the research, application of research, the adoption, the implementation, over to government policies under the aspect of technological innovation. Tornatzky and Fleischer (1990) identified the context and aspects of the decision making of the adoption and implementation as part of the process. Organizations do consider multiple factors in the technological context, organizational context, and external environment before adopting technologies. The TOE framework enabled me to provide answers to the contextual behaviors that influence the adoption behavior of new technologies in organizations.

The technological context relates to the internal and external aspects of the organization in the technology adoption process. Tornatzky and Fleischer (1990) defined the availability and characteristic as aspects of this context. While characteristics articulate the internal factors that the organization applies to the technology for the adoption and implementation, the availability aspect allows presenting factors suitable external technologies in the marketplace. Both aspects of external availability and internal characteristic of the technology allows an organization to decide the selection and adoption approaches for the new technology.

The organizational context relates to the aspect relevant to the organization. Tornatzky and Fleischer (1990) identify the communication process, the formal and informal linking structures, the size, and the slack of the organization as essential factors for the adoption of innovation. The communication process can affect the behavior in the adoption of technology as well as the structure of the organization in allowing to make decisions in the adoption process (Gangwar, Date, & Raoot, 2014). Awa, Ukoha, and Igwe (2017b) presented the factors slack and size of the organization as factors that might be stronger and or weaker in the organizational context. However, the existence of the organizational factors form an essential viewpoint in the TOE framework for the decision process of technology adoption in organizations.

The external task environment context relates to the influencing aspects of organizations in adopting innovations. Tornatzky and Fleischer (1990) explained the factors of industry characteristics, including the market structure, the support infrastructure, and the regulatory environment. Chandra and Kumar (2018) maintained the importance of the holistic view of the industry for adoption. Batubara, Ubacht, and Janssen (2018) recognize laws and regulations as an essential consideration in the process of adoption. The external environment of the operating organization will have beneficial or limiting aspects of any of these factors. Therefore, the strength in supporting the adoption decisions will vary.

These TOE aspects are essential for each organization to understand in detail the adoption of technological innovation. Gangwar et al. (2014) claimed that the TOE framework’s flexible inclusion of technological, organizational, and environmental factors, without being restricted to industries or company sizes, allowed researchers to establish a holistic picture of the adoption impact. Oliveira and Martins (2011) asserted TOE as a sound theoretical foundation for investigating the adoption of information systems innovations. Similarly, Hoti (2015) concluded the usefulness of the TOE framework for researchers and practitioners providing enabling insights into the adoption effect of a wide range of technology innovations. Gangwar et al. (2014) argued that the flexibility of the factors in the TOE framework does not allow generalization of the strategies for adopting an innovation. While the factors may vary between the organizations, each organization will require development of their set of strategies for the adoption of technology innovation. The TOE framework allows investigating the micro-level factors based on the influences of the macro-level of the context.

I used the TOE framework as the conceptual framework for my investigation of domain-driven design strategies and DevOps practices that IT product managers used to reduce the inefficiencies in the MSA adoption in organizations. The framework's structure was beneficial to identify influencing factors for adoption decisions from the data I collected. The factors supported to answer the research question and provided input for strategies and practices for the MSA adoption in organizations.

Technological Context. The technological context considers technologies that are relevant to the organization. Oliveira and Martins (2011) described the technological context as innovation that is not available in the organization as well as practices and technologies that are deployed and available within the organizational environment. Tornatzky and Fleischer (1990) defined innovation in the technological context by characteristics and availability. The definition is not very specific and provides the researcher with an opportunity to identify or to define the aspects of the innovation under a holistic viewpoint. The advantage of the broad definition is the increased possibility to cover all specific technological innovation factors under this context without modifying the framework.

Available technologies. Identification and selection of available technologies for the organization is part of the innovation process. Chandra and Kumar (2018) recognized that organizations should consider technologies in the marketplaces as well as internally deployed technologies to support the effective adoption of the technology. Wang, Li, Li, and Zhang (2016) identified that compatibility and complexity of the new technology when applied to an existing environment, impact the adoption within the organization. The introduction of new technologies into an organization requires, in many cases, tight integration with existing technologies to ensure efficient adoption and subsequent optimized operation. Therefore, organizations must make all efforts to select the best available technology that integrates with their current technology stack in use. Adopting innovations means a challenge for an organization that starts from the selection of the technology to the introduction into the organizational environment and the full corporate usage. The extent of the technology change to the existing technology landscape and the anticipated impact influences the decision to adopt the innovation.

Research on the innovation process and innovation characteristics are continuing as the new technologies are frequently emerging. Tornatzky and Klein (1982) reported as innovation characteristics relative advantage, complexity, communicability, divisibility, cost, profitability, compatibility, social approval, trialability, and observability as factors that influence the adoption decisions. Understanding the impact of technology change is essential for an organization to comprehend and to define appropriate actions for the integration into the current technology environment. Premkumar and Ramamurthy (1995) reported relative advantage, compatibility, and complexity as influencing factors in the adoption of inter-organizational information systems. Organizations even consider a small improvement to the business operations through the adoption of technology innovations as an advantage. Bhattacharya (2015) assessed that only relative advantage influences the adoption of radio frequency identification technology (RFID) in the technology context. The small improvement in the business process through tracking using RFID, enabling efficiency in the operational landscape of business. Danylenko (2018) presented four categories of innovations: (a) radical innovations of technical nature that result in new organizations and products, (b) recombination innovation by reusing existing solutions to define new products and technologies, (c) modification innovation of minor change to existing technologies, and (d) management innovations that change the organization or the industry. Categorizing the innovations into impact clusters of change dimensions allows organizations to merely assess the potential impact on the environment of the organization. Organizations try to understand in early stages to identify the usefulness and impact of the innovation on the business environment.

The technological context aspects are acting as a holistic umbrella for the adoption of innovation. Francesco, Malavolta, and Lago (2017) presented a categorization framework for microservices to identify essential criteria of MSA that are based on existing research contributions. Also, Cerny, Donahoo, and Trnka (2018) analyzed the existing research articles for microservices to identify factors that are relevant to describe the MSA. Both studies used the existing literature to identify factors relevant to describe the MSA and had similarities and differences of factors. Soldani, Tamburri, and Van Den Heuvel (2018) investigated the pains and gains of microservices and identified characteristics related to these aspects.

Similarly, Taibi and Lenarduzzi (2018) identified bad smell characteristics of MSA-based systems that impact the adoption of this technology. Both studies investigated aspects that have negative impacts on the adoption of MSA systems. While the findings of Taibi and Lenarduzzi (2018) were supported with values for harmfulness, Soldani et al. (2018) did not investigate the insensitivity of the pain and gain factors. Furthermore, some factors identified in these studies are similar, and others are entirely different and unique. In this study, I investigated the adoption of MSA under the focus of domain-driven design strategies and DevOps practices. I identified technological aspects in the form of characteristics and availability from the data collected. The openness of the TOE framework in the technological context allowed me to apply a holistic viewpoint in this research for the identification of relevant factors that influenced the adoption of MSA and strategies for domain-driven design and DevOps practices. However, the current equipment deployed in the organization is an asset that requires treatment and protection of the previous investment.

Current equipment and methods. Technology assets in the organization are an essential investment that supports business operations. Tornatzky and Fleischer (1990) asserted that internal technologies could play an influence on the adoption decision as a possible change of integration is needed. The equipment and configuration were an essential aspect for the adoption of new technologies as possible integration was required to improve the business operations. Awa et al. (2017b) reported for the technology context, and the factors perceived simplicity, compatibility, and performance expectancy as influences of adoption of information systems. Integration between the existing equipment and the new technology can become a complex undertaking as the possible experience of the impact and usefulness to business operations is unknown.

The integration of MSA into the existing IT landscape is crucial for the success of the adoption. Knoche and Hasselbring (2018) described the migration process of a monolithic software system to MSA-based system using a domain-driven design approach and refactoring of the software code. The adoption of MSA on an existing legacy system is a challenging process as ongoing integration with the monolithic application is required. Similarly, Soldani et al. (2018) identified the communication heterogeneity and application programming interface versioning for the integration with external systems as pain for the microservice adoption. Communication between the systems was an essential feature to be useful in the enterprise IT landscape to support the flow of information between legacy and MSA-based applications. Knoche and Hasselbring (2019) discovered compatibility issues, the maturity of technology, deployment complexity as a barrier for the microservice adoption. The adoption of MSA is a significant undertaking that requires a detailed strategy for design, development, implementation, integration, and migration. This study investigated case organizations that had large scale IT landscape in operations and required significant integrations with their MSA-based systems to support the business processes. I discovered domain-driven-design strategies that were useful to ease the integration effort. Furthermore, I identified DevOps practices that supported the migration and integration with the existing application landscape within the case organization. The variation of innovations and their technical characteristics resulted in variances of factors that influenced the adoption of the technology. While the characteristics depend on the technological context within the organization, the availability of the innovation with these characteristics on the marketplace might be crucial for the adoption.

Organizational Context. The organizational context covers the aspects of the organization's need for the adoption of innovation. Chandra and Kumar (2018) refer to the organizational context as features and resources available to support the introduction and operation of the new technology such as the structure and hierarchy, the communication process, the size of the organization, and the number of slack resources. While this context has two somehow measurable factors of size and slacks defined, the aspects of the communication process and structure are fuzzy to define for the adoption process. As businesses are differently built and operated, the context supports the researcher to include facets of the organization in adopting new technology.

Formal and informal linkage structure. The structure and hierarchy of an organization is an influencer in the process of introducing and operating new technologies in an organization. Tornatzky and Fleischer (1990) asserted that organizational structure influences the adoption process and recognizes the contextual constraints linked to the process and organization. Dekoulou and Trivellas (2017) claimed that supervision and training are two organizational structure dimensions influencing the innovation adoption in Greek's advertising and media industry. Daniel and Cooper (2017) reported that organizational formalization moderates the innovation behavior in Australian companies. Tornatzky and Fleischer recognized the influences of the structures and communication linkage for the adoption process in an organization as an essential component, that has been supported by the findings of recent studies of

Dekoulou and Trivellas (2017) and Daniel and Cooper (2017). Gemünden, Lehner, and Kock (2018) argued that the creation of project-based oriented innovation is influencing the adoption of innovation through project goals and value communication. A project has the advantage of a defined business context of the expected outcome, identification of resources required, risks identified, and a timeline that allows the project members to act within the organization. While the informal linkage and communication is an essential aspect in adopting technology innovation, the support from the leadership is needed for allowing the informal structures and exchange of information.

The adoption of MSA-based systems requires an organizational structure that supports the lifecycle of the services delivered. Callanan and Spillane (2016) described the structure of an organization adopting MSA using DevOps practices. Using a DevOps based structure simplifies the communication and software lifecycle management as teams are smaller than traditionally organized teams. Pallis, Trihinas, Tryfonos, and Dikaiakos (2018) argued that MSA-based systems are encouraging DevOps practice as structure and tool for the adoption. DevOps tools support the lifecycle of the MSA using automation in the process and limiting the dependencies to other software systems using domain-driven design strategies. Erich, Amrit, and Daneva (2017) reported that DevOps and MSA benefit from each other but DevOps do not require MSA as a engineering approach. DevOps improves software development and operational software management practices that benefit MSA-based systems due to the aspect created by domain-driven design approaches for the software. While DevOps can be applied to many software architectural styles, MSA-based systems seem to fit DevOps practices natively. The formal and informal structures of the case organizations provided insight into the DevOps practices to reduce the inefficiencies of MSA-based systems.

Communication and top management leadership behaviors. Top management leadership supports the collaborative effort in an organization. Tornatzky and Fleischer (1990) argued that clear goals communication, change communication, policies to reward innovation, enabling high skilled capabilities to foster the adoption of innovation in an organization. Hughes, Lee, Tian, Newman, and Legood (2018) asserted that the transformational behavior of the leader influences the support for adopting innovations that resulted from research among 100 top Iranian companies. Similarly, Sperber and Linder (2018) reported that suitability of the top management influences the strategies taken and the timelines in enabling the adoption in the organization. The principles of top management support for the adoption of new technologies identified by Tornatzky and Fleischer are maintained by the findings of recent researches by Hughes, Lee, Tian, Newman, and Legood (2018) and Sperber and Linder (2018). The support by the top management through the setting of goals, policies, communications, and establishment of capabilities is a long-term effort influencing the adoption process for technology innovations. Top management leadership behavior emerges in larger organizations to create an impact and drive the change into the organization.

Adopting MSA-based systems and DevOps practices results in a significant impact and change effort of the organization. MacLennan and Belle (2014) reported that top management supports the adoption of SOA by aligning the adoption strategy to the business strategy. While MSA is related to SOA, the involvement of top management is demonstrating a commitment by the organization in creating the environment of a successful adoption. Erich et al. (2017) concluded that leadership behavior is implicitly embedded in the DevOps practices as focused on the process of the software lifecycle. As DevOps is a cultural shift that involves new ways of working, strong support from the top leadership is required. The adoption of MSA and DevOps practices will require guidance from the top management for developing suitable strategies to adopt MSA-based systems. Otherwise, emerging challenges will be inefficiently addressed. My research investigated the strategies of domain-driven design and DevOps practices in the adoption process of MSA-based systems in the case organization that might be influenced by the top management leadership behavior.

Size of the organization. The size of the organization is considered to have more resources for skilled people, easier access to financial resources, established best practices, and other capabilities available to support the adoption of new technologies. Tornatzky and Fleischer (1990) recognized the difficulties of measuring the organizational size and determining the impact on the adoption behavior of the organization, using the size for exclusion purposes. Forés and Camisón (2016) investigated the innovation performance aspects using the size as a factor leading to different results for innovation adoption, radical innovation, knowledge creation, and incremental innovation. On the contrary, Martínez-Román and Romero (2017) reported a positive relation of firm size to technology adoption and innovation. While the size of the organization may have different influences in the adoption process, the complexity of the innovation adoption aspects is linked to aspects of the innovation adoption.

Researchers are investigating the relationship between the size of the organization to the adoption behavior of innovation. Leal-Rodríguez, Eldridge, Roldán, Leal-Millán, and Ortega-Gutiérrez (2015) argued that the size of the organization influences organizational change and performance in adopting innovations. Organizational learning is needed to revise and adopt new ways of working. Similarly, Dooley, Kenny, and Cronin (2016) reported that substantial size influences the effort in inter-organizational effort in adopting innovations is higher among small-medium to large-sized organizations compared to smaller organizations. Collaboration among organizations is typical for information technology innovations as best practices and customizations can be shared for the adoption process. Sharma and Rai (2015) reported that the size indicates a higher adoption rate recognizing higher available resources for the adoption of a computer-aided software engineering organizational (CASE). The adoption of sophisticated technology such as CASE requires to have a significant number of resources of people and proven best practices available. While the size of the organization influences the adoption process, it must be contextualized to the innovation type, technology, resources, and organizational aspects.

Adopting MSA-based software systems is an organizational decision to achieve organizational goals. I could not identify research that linked the size of the organization to the adoption of MSA. The organizational size discussion might be a consequence of the early research effort in microservices as other topics related to MSA are deemed more critical. In contrast, MacLennan and Belle (2014) reported that organizational size is not a critical factor in the adoption of SOA. Adopting SOA is an architectural decision for connecting software components, the impact having access to a large number of resources compared to a small, flexible organization is of less importance for the adoption. Zimmermann (2016) concluded that MSA are an organic approach to SOA with commonly shared characteristics. While SOA has a close relation to MSA, the organizational size influence has not been considered in research yet. The establishment of strategies for domain-driven design and DevOps as part of the MSA adoption may emerge as relevant to a part of a subset of a larger organization. However, I investigated organizations that had at least five development teams working on MSA-based system regardless of the size of the organization.

Slack. The availability of resources that understand the sophisticated technologies are incubators for the adoption decisions. Tornatzky and Fleischer (1990) presented slack as flexible resources that are available to the organization to influence the adoption process; however, not necessarily needed. The slack has a range of characteristics ranging from financial resources, knowledge, assets, to human resources that a technology adoption could require. Vanacker, Collewaert, and Zahra (2017) asserted that financial slack enhances performance while human resource slack lowers the performance of the organization. The motivation to adopt new technology is, in many cases, driven to improve or protect the performance of the organization. Suzuki (2018) concluded that the type of innovation is forming a constraint that influences the relationship between organizational slack and the innovation adoption process differently. The complexity of organizational structures and approaches to optimize the performance using frameworks, for example, Lean Management or Kanban, is making the identification of slack resources difficult. The decision of technology adoption is typically aligned with a detailed resource view and timeline to ensure a positive contribution to organizational performance.

The adoption of MSA requires a detailed plan and resource commitment to ensure the outcome supports the organizational goals. Zimmermann (2016) depicted nine practitioner questions that may be addressed in the adoption process for MSA and range from architectural decisions to organizational scaling strategies. Incorporating all the resource requirements in advance might not be possible and require addressing when emerging. Soldani et al. (2018) analyzed and reported different issues in the adoption process of MSA ranging from design, development to operational topics. The adoption of MSA required flexible and adaptable resources and contextual knowledge to ensure that the problems did not impact the outcome negatively. While I investigated the strategies for domain-driven design and DevOps practices, the available slack resources at the case organizations did not influence the approach for the strategies to deal with the emerging issues during the adoption of MSA.

External Task Environment. The external task environment presents the characteristics of the industry, the availability of support for the new technology, and the operating environment of the organization. Tornatzky and Fleischer (1990) argued that the adoption of innovation is influenced by external aspects of competitiveness of the industry, the existence of supporting infrastructure for the new technology, and the regulatory environment. While achieving competitiveness is a strong motivation of a company to improve and adopt new ways of working, mitigating risks by using an external support environment is ensuring that the change is contributing to the success of the company. The operating environment of the organization is constraint by the regulatory requirements that influence the organization in their decision making for the adoption of new technologies. These three aspects are influencing the adoption of innovation from an external viewpoint.

Industry characteristics and market structure. The competitive pressure and shifting customer demands increase the need to innovate using new technologies to sustain the performance level of the organization. Hashmi and Van Biesebroeck (2016) reported for the automotive sector that companies have higher adoption rates when operating in a market with a broad range of quality levels and reducing innovations in highly competitive environments. Continually observing the market and carefully selecting a new technology before adopting seems to fit the nature of organizations in competitive markets, while organizations with a wide range of segments need to differentiate with innovations. On the contrary, He (2015) reported for U.S mobile banking sector high technology adoption decisions in concentrated market structure and least innovation adoption in highly competitive markets. While in emerging markets, a low rate of innovation adoptions creates the differences, concentrated markets require the organization to use technologies to differentiate on non-price aspects. Gottinger (2016) argued that little agreement exists in the science community for the relationship between innovation and intensity of competition as tricky to understand the linkage between both. The complexity of market dynamics and the response of organizations to address these conditions influencing the decisions for the adoption of new technologies. Kohli and

Melville (2019) explained the alignment of actions to the external competitive environment of managers with internal capabilities during the adoption of digital innovations in the organization. The synchronization of the internal resources to adopt an innovation is a vital approach to justify the actions and creating the internal best practices for technology adoption. Investigations into the best practices for the adoption of innovation will require to understand the organization's external environment and market behaviors in detail.

The ability of an organization to adapt to new market conditions within a defined timeframe is essential for its survival. Jamshidi et al. (2018) depicted microservice architecture to increase the agility of software systems in aligning better with business requirements. Delivering new software functions quickly into the business environment ´-by using domain-driven design and DevOps for MSA-based applications is one of the aspects of adopting MSA-based systems. Also, Knoche and Hasselbring (2019) reported the results of a survey among professionals in Germany as the primary reason for adopting microservices scalability, maintainability, and time to market. The scalability aspect of microservices supports the business in allowing to start small and scale quickly once the business demands increases. Williams, Sica, Killen, and Balis (2016) asserted to adopt microservices in the bioinformatics area to increase the collaboration and lower the software maintenance effort for these systems. The available architecture pattern for microservices and simple integration of other microservices to establish a software system is vital in the research and development area to try and test new ideas. MSA is seen as an enabler for further innovations as it allows to quickly introduce new systems that work across internal organizational and external service boundaries. This study investigated the strategies for domain-driven design and DevOps practices needed for the adoption of MSA that were not influenced by the market structure and market condition of the organization operating. The aspect of the industry characteristics might influence the way in designing the MSA-based applications as requirements demanding a specific microservice construct. The market structure might influence both the domain-driven design strategies and DevOps practices as specific ways of working demand an alignment with the external business environment.

Technology support infrastructure. The supporting infrastructure for new technology influences the adoption of innovation in organizations. Tornatzky and Fleischer (1990) presented the technology support infrastructure as depended on the labor cost, skills of available resources, and the access to technology suppliers. Having the right skills at the right prices is a major concern when adopting new technology, enabling support from suppliers that have done similar technology adoptions to apply best practices to reduce the risks for failure is another aspect for the adoption. Martínez-Román and Romero (2017) observed that services from external consultants influence the adoption of technology. Many consulting organizations have standardized offerings around technology adoption and enhancements providing best practice knowledge and experienced resources to companies on demand. Ramanathan and Krishnan (2015) argued that the availability of support for the enterprise influences the adoption decision of open-source software. The introduction of a complex product such as a piece of software into an organization requires knowledge to maintain and enhance to ensure availability for the business operations when needed. The dependency of access to a technology support infrastructure will vary for each technology type that the organization intends to adopt. While the access to skills and labor cost is one driver, the availability of technology providers is another. Organizations must decide the best strategy of using recruitment to onboard available skilled resources and a sourcing strategy to enable access on-demand to the technology supplier.

Skill availability and access to technical support is a vital consideration for the adoption of MSA systems. Knoche and Hasselbring (2019) reported insufficient skills for developers and operating staff as the most significant hurdles for the microservice adoption. The complexity of a distributed system developed in multiple-software languages with eventual consistency of data is high and requires an in-depth knowledge of technology and broad expertise in development and operation. Soldani et al. (2018) identified pains of microservices development for identification of the bounded context and ensuring consistency of data and microservice operations the distributed and dynamic behavior with the increase of resource consumption. Also, Taibi and Lenarduzzi (2018) concluded that microservice smell could be a challenge for the developers and operational personnel because of the architectural design decisions, complexity in maintenance, distributed systems, and operational issues with multiple connectivity endpoints of the services. The handling of complex systems requires significant knowledge and best practices to ensure efficient development and operations. While domain-driven-design strategies and DevOps practices can be of help in the development and operations of MSA systems, appropriate practices for the overall MSA-based will be required. These practices might be able to come from a specialized consulting organization or acquired by hiring these expert skills from the market.

Government regulation. The regulatory environment of the organization influences the adoption decisions of new technology. Blind, Petersen, and Riillo (2017) discovered that regulations influence the adoption of innovation positively and formal standard influences the adoption negatively in low, uncertain markets and is reversed in highly uncertain markets. Typically, formal standards are helping to establish conditions that allow to share and exchange technology innovations in higher volumes. Regulations tend to create equal conditions for innovations in organizations. Stern (2017) reported an additional 7.2 months approval time for a new medical device type under the U.S. regulations influencing the decision to adopt such technology. Medical regulations are focusing on the risk and require significant resources for the adoption to ensure no adverse effects in using the technology emerges. Zewen, Xin, and Hongjun (2017) asserted for regulations in green innovation policies a significant and positive impact on green technology adoption in organizations. Policies for environmental efficiencies support the adoption of technologies to ensure compliance their lack of achieving can have a negative impact on the reputation of the organization and performance.

Similarly, Bossle, Dutra De Barcellos, Vieira, and Sauvée (2016) observed regulations as the most influential factor in adopting green technologies in organizations driven by compliance with standards. Adopting green technologies supports the drive to reduce waste in processes and resources, leading to cost savings with performance improvements of the organization. While policies play a role in establishing equal market conditions for innovations, formal standards ensure the inter-organizational exchange and adoption of technology innovations.

Regulations and formal standards influence the best practices and strategies for software development and operations of the software systems. Knoche and Hasselbring (2019) reported for microservice adoption the barrier of compliance and regulations varies between the industries with financial services as the highest score of 1.50 and the lowest score of 0.45 in the energy & industry. The importance of regulations in specific industries is influencing the adoption behavior of new software architecture systems significantly as the additional effort is required to ensure compliance with these regulations. Laukkarinen, Kuusinen, and Mikkonen (2018) illustrated the approach to adopt DevOps practices in highly regulated industries for medical device and health software development standards of IEC 63204 and IEC 82304. Typically, regulatory requirements demand addressing the tracing of requirements to code, documentation of the development, repeatability of the test cases, and secure code deployments that should be auditable for authorities. Rademacher et al. (2018) explained the bounded context of the domain-driven design links to an isolated business capability mapped to a microservice covering all functionalities of the software services. The bounded context is the outcome of the design to ensure all needed functionalities are addressed from a business perspective, including the regulatory requirements. While the domain-driven design covers the business functionalities of the MSA-based system, DevOps practices enable the regulatory compliant life-cycle handling of the MSA-based systems. The outcome of the domain-driven design is an application construct of a bounded context per

MSA system that is directly aligned to the business and regulatory requirements. The DevOps practices should comply with the regulatory requirements over the lifecycle management that might have directly or indirectly impact on MSA-based systems and DevOps practices. The influence of the regulatory requirements on the domain-driven design strategies and DevOps practices related to the adoption of MSA-based systems was not visible for the two case organizations of my study. While in the global consumer goods manufacturing industry, the focus of the regulation is on the development of the physical product and reliability of the functionality, the focus of regulations in the financial industry is on the risk exposure of the induvial financial performance. The regulations applicable to the MSA-based system influenced both the domain-driven design strategies and DevOps practices.

Analysis of Related Theories

Several theories exist in information technology about technology adoption. Awa et al. (2017a) identified two groups of technology adoption theories, one at an individual level and one at the organizational level. On the contrary, Gangwar et al. (2014) depict three adoption levels with the individual, group/team, and organization. Theories that include individual behavior are, for example, the technology acceptance model (TAM) developed by Davis (1986), the theory of planned behavior (TPB) developed by Ajzen (1991), unified theory of acceptance and use of technology (UTAUT) developed by Venkatesh, Morris, Davis, and Davis (2003). Theories at the organizational level are, for example are the diffusion of innovation (DOI) developed by Rogers (1983), the technology organization environment (TOE) developed by Tornatzky and Fleischer

(1990), and the decision maker-technology-organization-environment (DTOE) developed by Thong (1999). While the adoption behavior on the individual level targets the acceptable behavior on single technology interactions, the adoption behavior on an organizational level is related to the factors influencing the adoption of technology within the organization. Microservices are software systems that are developed and maintained by organizations that support their operational needs. Microservices can be exposed to other systems or applications in front ending user applications. The technology adoption on an individual level could provide insights for organizational adoption decisions.

Technology acceptance model. The TAM theory considers the usefulness and ease of use as an influencer of technology adoption. Davis (1986) developed TAM to understand the acceptance of computer-based information systems and predict the success of the adoption of such a system. In 1989, Davis defined a revision of the TAM theory to support a broader software technology scope and user base. The understanding of essential aspects of a user's adoption approach for a software system provides input on the development and the expected benefits. Tarhini, Arachchilage, Masa'deh, and Abbasi (2015) presented TAM as a framework with five distinctive aspects of (a) attitude towards behavior related to steering the adoption behavior, (b) behavioral intention measuring the strength of the adoption behavior, (c) perceived usefulness (PU) evaluating the performance of the technology, (d) perceived ease of use (PEOU) describing the user's positive experience of the technology, and (e) external variables include factors that influence PU and PEOU. Relying on the usefulness, ease of use, and influencing aspects from external allows researchers to investigate the technology adoption from a user's perspective. While researchers can include specific technology factors or contextual aspects, the TAM theory does not suit every research topic.

The evolution of TAM presents new opportunities and challenges for the researcher using the theory. Lai (2017) recognized that the enhancements of TAM are providing new insights into the adoption situation of the user over time and in complex IT environments. The motivation of adoption of technology can variate over time; for example, the experience of the user increases for the adopted technology, new requirements may be established, or other technologies emerge that compete against existing technology. Koul and Eydgahi (2017) presented the extension of TAM to overcome the user's adoption behavior and social influences. The easy access for all generations to technologies and their social environment influences the individual to adopt new technology. Also, Evwiekpaefe, Chiemeke, and Haruna (2018) argued that TAM is suitable for a broad range of technologies in multiple research situations and user groups. The proven flexibility of a framework in supporting multiple technologies allows the researcher to apply it to complex research settings. While TAM has been tested for the behavior of technology adoption of a user, it does not explain how an organization adopts new technologies.

The TAM theory explains the technology adoption of organizational users and their influencing factors. Ehteshami (2017) investigated the barcode acceptance in hospitals using TAM with a focus on the PU and POUE extracting recommendations from the survey analysis. TAM is useful to understand the user's motivation and barriers in adopting technology to develop a possible, desirable course of action. Similarly, Folkinshteyn and Lennon (2016) were grouping Bitcoin users into developers and end-users and analyzed the effectiveness with TAM to identify possible actions for each user group. New technologies such as blockchain and Bitcoin present a sophisticated environment for the developer and end-user to handle, with each in need of customized user experience to successfully adopt the technology. Steininger and Stiglbauer (2015) modified the TAM using PU only for the electronic health record (EHR) investigation to understand the possible implication for a new EHR system that is not deployed yet. The focus on the user to understand the implications for technology adoption allows developing strategies for the system on an organizational level. This study investigated the strategies for domain-driven design and DevOps practices used by IT product owners during the MSA adoption. The domain-driven design is an approach to design the MSA-based systems using DevOps is a set of practices from the development into operations. The organization is defining the strategies for the MSA adoption of the organization to improve the business operations and not to improve user adoption behavior. Therefore, I did not consider TAM as a framework for my study as the focus is on the organization level adoption of MSA-based systems.

Diffusion of innovations. The DOI theory considers stages of innovation adoption at an organizational and individual level. Rahayu and Day (2015) described that the initial DOI model was developed by Rogers in 1983. In the fifth revision of the Diffusion of innovation book, Rogers (2003) argued that the adoption of innovation is a communication process via channels of a social network, called diffusion, that occurs over time. The dissemination of ideas in an organization can be embedded in a process that improves the operations of the business or resulting from the transformational requirements of the business. Rogers (2003) developed five adoption categories of innovators, early adopters, early majority, later majority, and laggards that have a different extent of adoption time. The scale of the distributions of technology innovations and their adoption in organizations will change once success reports and best practices are made being available to a broader community. Rogers (2003) defined three contextual characteristics (a) individual (leader) characteristics of attitude towards change; (b) internal characteristics of organizational structure with factors of centralization, complexity, formalization, interconnectedness, organizational slack, and size; (c) external characteristics of the organization as of system openness as factor describing the organizational innovativeness. The technology adoption approach in organizations is a complex undertaking, and multiple individuals, rules, and regulations influence the decision process. While the framework articulates the variation of decision making over stages, it is not without weaknesses in explaining the adoption process.

The diffusion of innovation is a generic framework for innovation adoption, missing a level of clarity. Atkin, Hunt, and Lin (2015) argued that complex technology innovations such as new media are complex and not fully covered by the DOI theory. New technology innovations such as MSA-based systems comprise in many cases of a set of technology clusters compared to a single innovation. Also, Tarhini et al. (2015) asserted that the DOI model misses a clear linkage between the innovation adoption impacting the application in the research. Extension of the DOI framework by including new context characteristics or new factors is a simple way for the researcher to enhance the applicability as a framework for research. Evwiekpaefe et al. (2018) recommended enhancing the DOI theory with the environmental context and decision attitude being relevant for technology adoption. The extension of DOI with environmental characteristics will improve the completeness for an investigation into technology adoptions. This study investigated the MSA adoption and their strategies for domain-driven design and DevOps practices. MSA-based systems consist of multiple innovations that require adoption by an organization. While I could enhance the DOI framework with environmental characteristics, the complexity of the MSA adoption is not addressed appropriately. Additionally, I did not intend to investigate the individual's attitude as interested in the organizational strategies for domain-driven design and DevOps practices. Therefore, I did not consider the DOI theory as suitable for my research.

Usage of TOE in the Research

Researchers consider TOE theory a sound theoretical model for their investigations. Senyo, Addae, and Boateng (2018) reported for the cloud computing research between 2009 to 2015, that the TOE theory is the most used research model with 5.3 percentage as standalone and 1.2 percent in combinations of TAM and TOE, TOE and Organization-Technology–Fit (HOT-fit), and TOE and DOI frameworks. Similarly, Niknejad et al. (2018) concluded for the service orientated architecture adoption research between 2009 to 2017, that the TOE framework is the dominant research model. While in both studies, most researchers did not use a framework, the TOE is for many researchers a sound choice for adoption research. The use of a proven theory supports the effort of the researchers in investigating the adoption of new and emerging technologies.

Quantitative research design. TOE theory is the dominant choice for researchers in the adoption research. Several researchers use TOE as a theoretical base for the investigation (Ahmadi, Nilashi, & Ibrahim, 2015; Hsu, Ray, & Li-Hsieh, 2014; Lian, Yen, & Wang, 2014; Senyo, Effah, & Addae, 2016; Srivastava & Nanath, 2017; Yang, Sun, Zhang, & Wang, 2015). The TOE framework allows the researchers to define relevant factors for the three contexts to establish a broad enough technology adoption definition with the appropriate level of details for the investigation. Senyo et al. (2016) reported the findings of a survey from 305 organizations in Ghana for the cloud adoption in a developing country based on the TOE framework. The three contexts allowed the researcher to broaden the viewpoints for the technological adoption identifying influential factors that may exist in the organizational and external task environment. Also, Srivastava and Nanath (2017) analyzed the cloud adoption barriers in UAE with the focus on external factors of a strong cloud ecosystem support using a survey from 25 organizations for data collection.

The researcher has the flexibility to use the TOE framework to structure the findings collected from a survey under the three contexts to present actionable recommendations for the reader of the study. Hsu et al. (2014) investigated the cloud computing adoption empirically with factors that included pricing and deployment to explain the adoption decisions. Also, Yang et al. (2015) developed a tripod model of Software as a Service (SaaS) readiness employing TOE theory and a survey collecting 173 responses linking the three contexts as influential for the SaaS adoption. TOE theory offers a sound empirically tested framework for the researcher to understand the influences of the various contextual factors for the adoption of the innovation. MacLennan and Belle (2014) investigated the organizational adoption of service-oriented architecture (SOA) in South Africa. MacLennan and Belle (2014) defined the contextual factors by examining factors of six previous studies of web services adoption, including hypotheses testing for the stages of use of SOA and project success. MacLennan and Belle (2014) reported significant differences in factors between the stages of SOA adoption. The TOE framework enabled the researchers to categorize possible factors influencing the adoption of SOA and empirically validate the significance in specific phases of the adoption. While TOE allows the researcher to cover many aspects on an organizational level, it does not establish a specific investigation focus.

Enhancing the TOE with additional context enables additional coverage of a researcher's introduced context in the adoption investigation to establish an enhanced focus. Lian et al. (2014) examined the critical factors for the cloud adoption in Taiwan hospitals by enhancing the TOE framework with Human-Organization-Technology (HOT)-fit model to include the human dimension in the research. Similarly, Ahmadi et al. (2015) investigated the hospital information system adoption of Malaysian public hospitals merging TOE and HOT-fit theory and using for the data analyzes a hybrid multi-criteria decision-making (MCDM), analytic network process (ANP) and decision-making trial and evaluation laboratory (DEMATEL) approach. The importance of including the human factor in healthcare-related research is essential as a human is the primary concern of all activities. While much of the TOE theory-based research is quantitatively based, the researcher can choose other designs for a TOE theory-based investigation as well.

Qualitative research design. Researchers use qualitative methods of investigations to gain an in-depth understanding of the adoption considerations. Several researchers combined TOE theory with various qualitative research designs such as case study (Al-Hujran, Al-Lozi, Al-Debei, & Maqableh, 2018; Bradford et al., 2014; Kurnia et al., 2015; D. Leung, Lo, Fong, & Law, 2015). The use of TOE theory with qualitative research allowing the researchers to develop explorational approaches to the adoption of gaining in-depth insight aspects. Leung et al. (2015) used a case study design to investigate the initial adoption and continued adoption of information and communication technology (ICT) of an independent hotel in Hong Kong. Leung et al. identified the factor for each context of TOE from previous studies. Leung et al. employed semistructured interviews for the data collection and mapped the codes to the factors of the TOE framework resulting in a framework of factors that described the initial adoption and continued adoption. The use of TOE allowed the researchers to enhance the adoption framework with identified factors from the data analysis. Al-Hujran et al. (2018) analyzed the challenges of cloud computing adoption (CCA) by organizations in Jordan using interviews for data collections and detailing the results on the technological, organizational, and environmental contexts. Al-Hujran et al. (2018) reported the identified challenges under each context, establishing a TOE-based CCA framework with identified factors. The flexibility of the TOE framework allows researchers to group the qualitative identified findings under the various TOE contexts to describe possible implications of technology adoption. While TOE allows to modify and enhance the defined context with factors, other investigations frameworks cover other specific innovation aspects better.

Enhancing TOE with other frameworks improves the investigation approach for particular innovation technologies. Kurnia et al. (2015) analyzed the business-to-business electronic commerce (EC) technologies adoption employing a multi-case study design with TOE theory as a leading research framework to guide the investigation of influential adoption factors. Kurnia et al. (2015) included diffusion of innovation (DOI), resource dependence theory (RDT), and national culture theory (NCT) to complement TOE and gain an in-depth understanding of the influential EC adoption factors involving eight organizations within the Indonesian grocery industry. Kurnia et al. (2015) reported the findings in the TOE structure, including additional factors identified that were relevant in the adoption of EC. The enhancement of TOE with other theoretical lenses allowed the researchers to increase the level of detail for the investigation.

Similarly, Bradford et al. (2014) employed the TOE framework to a multiple-case study approach investigating the challenges of centralized end-to-end identity and access management (CIAM) and ERP systems. Bradford et al. (2014) selected two case organizations with 19 participants for interviews using three relevant questions regarding constraints, benefits, and effectiveness of CIAM and ERP. Bradford et al. (2014) reported the findings within the TOE contexts and enabling the reader to extract relevant considerations for best practices and policies. Using TOE with multiple-case study design supports the researchers in gaining an in-depth understanding of the factors in adopting a complex technology system extending the scope beyond the technology implications. While Kurnia et al. (2015) reported the findings in line with the factors of the TOE, Bradford et al. (2014) reported the findings without the linkage to any factors within the TOE context. While Kurnia et al. (2015) established the TOE factors for EC adoption from a previous research paper, Bradford et al. (2014) did not rely on factors from previous TOE-based CIAM and ERP studies. The investigation using TOE as a guidance structure in multiple-case studies enabled the researchers to present the findings with detailed and thick descriptions for a reader to understand possible applications in a different organizational context.

Critics of TOE

Every theory has limitations that impact the application in research situations, making the TOE theory no exception. Awa et al. (2017a) argued that the TOE theory is a generic model without the factors to explain the specific technology requirements. Similarly, Gangwar et al. (2014) emphasized that TOE does not have an explicit internal construct and factors defined in each context. Also, Sun, Cegielski, Jia, and Hall (2018) depicted TOE as an overarching theoretical model that required additional theories to address complex technology adoption situations. The missing clarity of the TOE framework creates difficulties in explaining the variations of findings for the adoption of innovations. While the TOE theory has been developed to act as a generic technology adoption model, the knowledge of innovation adoption has increased.

The TOE factors are a significant concern for the researcher utilizing the generic adoption framework. Hoti (2015) recognized that the researchers report different factors that influence technology innovation adoption, including factors that could not be integrated. Similarly, Chandra and Kumar (2018) determined the contextual factors of the TOE framework for the adoption of augmented reality in e-commerce based on the technology under investigation and previous research findings, for example, including technology competence or decision-makers' knowledge. Also, Wang et al. (2016) proposed factors for the adoption investigation of a mobile reservation system utilizing previous studies. Empirical research requires well-defined variables to enable data analysis for a result that must be established for the TOE framework. While the determination of the TOC factors is a concern, it presents an opportunity for the researcher as well.

Researchers have an opportunity to define suitable factors and enhance the TOE context. Ilin, Ivetić, and Simić (2017) proposed to combine DOI with TOE for the ERP investigation as both overlaps for organizational and technological context but enhances individual characteristics and environmental aspects. Also, Awa et al. (2017b) justified the extension of TOE with the UTAUT framework as not sufficient for a research context as individual and task context is missing. Martins, Oliveira, and Thomas (2016) defended the combination of TOE theory, DOI theory, and Institutional (INT) theory with the weaknesses of TOE for the factors of SaaS. Martins et al. (2016) argued that the TOE does not consider other factors such as cost or security for the adoption of SaaS and does fully support an explanatory research design. While DOI supports a deeper understanding of the adoption decision in the technological context, the INT explains the constraints of social and cultural factors of the environment context (Martins et al., 2016). The combinations of multiple theories with TOE allow researchers to address the weaknesses of the explanatory research design of the TOE. However, the flexible inclusion of other theories and factors for the TOE context is considered by researchers as a weakness and advantage at the same time.

Microservice Architecture System Adoption

MSA adoption has gained significant momentum as organizations increase their digitalization of services and products to ensure relevance in the marketplace. Taibi and Lenarduzzi (2018) referred to organizations of Amazon, Netflix, LinkedIn, and SoundCloud that have MSA adopted to enable independent development and deployment of services. Pallis et al. (2018) described the MSA adoption of Netflix, Amazon, and Uber due to the implementation of best practices in these organizations using DevOps for the management of the software lifecycle. The implementation of best practices is an evolving organizational aspect enforced by the drive to improve the quality of the outcome. Knoche and Hasselbring (2019) argued that companies are publishing MSA knowledge and tools either as a blog or as open-source software enabling other organizations to adopt the same quickly. The free access to knowledge and resources supports practitioners in their tasks to solve problems increasing the adoption of MSA. However, Francesco et al. (2019) reported that scientific research has increased between 2015 to 2017 for MSA with the majority of publications as conference papers and few journal papers. While the practitioners apply the learning and knowledge from others in their context, researchers require a common ground to ensure new knowledge can be established.

Definition and categorization of MSA in research. The academic research community is slowly overcoming the difficulties of the definition and categorization of the MSA as several practitioners presented definitions based on their experiences resulting that researchers struggle to define the scope of the phenomenon. According to Zimmermann (2016), the term and definition originated from online sources starting in 2014 with Lewis and Fowler (2014) establishing the terminology of microservices describing nine characteristics for MSA: (a) components based on services, (b) aligned to business capabilities, (c) based on a product, (d) smart interfaces and simple exchange structure, (e) operating independently, (f) distributed data organization, (g) automation of infrastructure, (h) build in failure handling, and (i) anticipate a change of design. The characteristics listed by Lewis and Fowler (2014) include MSA aspects, the external architectural requirements, and operational conditions to run MSA-based systems. In contrast, Zhu, Bass, and Champlin-Scharff (2016) referenced the definition established by Newman (2015) that presented seven principles related to microservices: (j) hide internal implementation details, (k) model around business concepts, (l) decentralize all the things, (m) adopt a culture of automation, (n) independently deployable, (o) isolate failure, and (p) highly observable. Newman’s principles articulated architectural aspects, operational considerations, and processes recommendations. Similarly, Zimmermann (2016) argued that both definitions are mixing the organizational process, software development concerns, and software architectural considerations that impact the clarity of the research community. Practitioners do tend to include everything relevant for the application to real-world problems. While some researchers might struggle to categorize and define MSA, others have started to apply similar complex definitions to other emerging architectural concepts.

Cloud-native applications are another architectural concept with a complex definition and linked to MSA-based systems. Kratzke and Quint (2017) combined both characteristics and principles in their definition for cloud-native applications (CNA), including mentioning microservices as composition approach. The inclusion of another complex concept of MSA in a definition is providing a more straightforward definition for CAN. However, it does not create a better understanding of the concept. Similarly, Thönes (2015) described the microservices based on characteristics articulating the responsibilities and complexity as of (q) small application, (r) independently deployed, (s) tested independently, (t) scaled independently, and (v) single responsibility. The definition by Thönes (2015) does not include any definitions for external requirements such as automation or anticipate changes. Practitioners tried to articulate the aspects that matter most to address the challenges of businesses and organizations. A recent definition for MSA by Knoche and Hasselbring (2019) combined architectural, organizational, and external characteristics as a baseline for the investigation into drivers and barriers of MSA adoption. Francesco et al. (2019) reported that the majority of researchers refer to the MSA definitions by Lewis and Fowler (2014) and Newman (2015). Falling back on the first definition allows the researcher to value the previous knowledge established for the investigation and avoiding a deviation into semantical topics of the research focus. The definitions for MSA include many aspects, including architectural concerns that enable to define the scope and develop the design of the MSA, considerations to orchestrate the software development, and operational activities in the most efficient form using concepts such as domain-driven design and DevOps practices. As these considerations are itself complex concepts, researchers struggle to isolate the MSA definition from other concepts to provide a platform for their research focus. Nevertheless, the adoption of MSA is continuing as anticipated benefits outweigh the challenges for the organizations to provide services based on MSA-based software systems.

MSA adoption research. An organization's decisions to adopt new technology is based on an assessment of benefits and challenges expected by the latest technology contributing to the organizational goals. Williams et al. (2016) presented the increase in productivity as the underlying motivation for the use of MSA in the application design by an organization such as General Electric, Amazon, Netflix, and Hewlett Packard. Improving productivity is vital for organizations with mature processes for development and operations. Similarly, Balalaie et al. (2016) argued that MSA and DevOps had been established as best practices from software vendors and content providers as DevOps enabling practical implementation of MSA. While DevOps can support the development and operations of other software architecture constructs, MSA seems to be a natural fit with DevOps. Also, Baškarada, Nguyen, and Koronios (2018) argued that the complexity of managing MSA-based systems requires DevOps practices to optimize the effort in the SDLC. Efficiency in the SLDC is crucial for organizations to control the resource requirements. While one organization is focusing on improving the quality, others are focusing on improving the efficiency in the software development system lifecycle

(SDLC). Typically, the large organization uses best practices for the introduction of changes to the existing business operations to ensure the change delivers the benefits anticipated.

The existence of best practices in other organizations provide the input for the realization of the benefits and controlling of the challenges in adopting new technology in organizations. Soldani et al. (2018) concluded that the industrial adoption of MSA had reached some degree of maturity, while the academic research understanding of the MSA phenomenon lacks behind. The expected benefits fuel the speed of industrial adoption; however, the challenges of adopting MSA in organizations are significant.

Understanding the benefits and challenges is crucial for organizations to define the course of action for the adoption of new technology. Baškarada et al. (2018) reported ten main challenges and two opportunities for interviewing 19 architects on the challenges and opportunities with the adoption of MSA. The challenges were (a) lack of relevant skills for the development and operations of distributed systems (b) the use of software as a service (SaaS) and commercial of the shelf (COTS) products in IT landscape impacts the use of MSA; (c) organizational culture in embracing the changes towards small development teams; (d) governance changes for distributed systems differs significantly from traditional frameworks and processes as impacting other systems; (e) organizational structure shifting from silos of a plan, build, run to product-oriented structures ensuring end-to-end responsibility of plan to run; (f) decomposition an existing monolith is a difficult task due to refactoring activities before splitting in microservices; (g) distributed master data management and consistency of data set is a major concern;

(h) event-based requires service choreography as the orchestration is not useful; (i) complex testing of distributed architecture systems catering all possible failure situations; (j) performance penalties of MSA-based systems (Baškarada et al., 2018). The opportunities were (k) increase of agility for software development and simpler deployment of the MSA-based systems; (l) simpler scalability of MSA-based compared to monolithic applications (Baškarada et al., 2018).

While adopting of MSA is not a simple task in addressing the challenges, organizations see significant benefits by delivering new functionalities faster with the possibility to adjust the performance depended on business demand. Also, Knoche and Hasselbring (2019) investigated the drivers and barriers for MSA adoption in Germany using a survey with 71 respondents. The primary drivers were identified as (m) scalability, (n) maintainability, and (o) time to market, including secondary drivers as (p) enabler of DevOps and continuous delivery, (q) suitable for cloud, and (r) supporting organizational improvements (Knoche & Hasselbring, 2019). The significant barriers were identified as (s) insufficient operational skill, (t) insufficient development skill, (u) resistance of operations, (v) consistent backup, (w) compliance and regulations, and (x) deployment complexity (Knoche & Hasselbring, 2019). Changing technology requires people with the right skills for development and operations. While the finding for the benefits aligns between Baškarada et al. (2018) and Knoche and Hasselbring (2019) for scalability, maintainability, and time to market, Baškarada et al. (2018) did not identify any secondary motivations for adopting MSA. The difference might be caused by the two different research approaches of interviews vs. a survey with predefined options in the questions. Similarly, the challenges between Baškarada et al. (2018) and Knoche and Hasselbring (2019) align for skills in development and operations, including the organizational resistance but differentiate from Knoche and Hasselbring (2019) of the remaining challenges. As Knoche and Hasselbring (2019) established the barriers based on personal experience, the differences to Baškarada et al. (2018) are significant. However, the challenges can be grouped into domain-driven design topics (a), (b), (f), (g), (j) and DevOps practices issues (c), (e), (h), (s), (t), (u), (v), (w), and (x). Similarly, the benefits can be grouped into domain-driven design topics (j), (l), (m), (n), (o), and DevOps practices (p), (q), and (r). The challenges might vary in importance for each organization but must be addressed to ensure the benefits of MSA adoption can be realized. However, both studies do not present any strategies to address the identified challenges neither recommend best practices for the domain-driven design nor DevOps in the MSA adoption approach.

Identification of bad smell can act as an indicator for generating problems or quality issues for software systems. Taibi and Lenarduzzi (2018) identified a catalog of 11 MSA bad smells using a survey approach with 72 participants of practitioner events between 2016 and 2017. The description of the bad practices included the impact and adopted solutions by the practitioners that could serve as input for the strategies of domain-driven design. The extracted strategies are (a) application of semantical versioning of application programming interface (API) to ensure correct communication between the microservices, (b) introduce patterns of API gateways to contain cyclic calls between microservices, (c) apply lightweight message bus for MSA-based communication and avoid an enterprise service bus (ESB), (d) adopt service discovery instead of hard-coded endpoints for services to remove location dependency, (e) avoid to request internal data of other microservices and revisit coupling aspects to ensure data consistency, (f) apply the right size for the microservice approach by creating only needed microservices to support maintainability, (g) use API gateways for simplifying MSA-based communications to avoid communication issues, (h) avoid or minimize the use of shared libraries for the microservices to ensure dependency of the microservices, (i) adopt data ownership for each microservice to ensure service independency by not sharing same data entities, (j) minimize the use standards to support appropriate knowledge and maintainability, (k) focus on MSA-based systems aligned to business capabilities to reduce the complexity of data management (Taibi & Lenarduzzi, 2018). The identified smells are a compilation of the most commonly reported problems by the practitioners related to architectural decisions for microservices. Taibi and Lenarduzzi (2018) measured the harmfulness of the smell by a 10-point Likert scale rating of the practitioners with rightsizing as most harmful, followed by hard-coded endpoints and not sharing the same data. The rating of the harmfulness provides a subjective perception of the impact and does not provide a view of the level of impact on the organization. The understanding of the implications is essential for an organization to define the appropriate actions and practices to contain or avoid the problem. The identified bad smells and presented solutions are aspects of the architectural implications; they do not indicate the impact on organizational practices or the organization itself. Hence, strategies will be required to address the domain-driven design approach and DevOps practices for the adoption of MSA-based systems in organizations to ensure the benefits are delivered by the systems. While organizations are concerned with the benefits and challenges in adopting MSA-based software systems, researchers require a common understanding of the phenomenon to ensure the improvement of knowledge in MSA.

Service architecture. The existence of a relationship between MSA and SOA is an ongoing debate between practitioners and researchers that will not end soon. Moskalenko and Berezenko (2017) defined SOA as a technology and product independent approach to enable simple, functional enhancement and capacity expansion of a software-based service. The definition by Moskalenko and Berezenko (2017) is too simple and does not explain the architectural decisions that make a service-orientated architecture different from other architectural standards. Cerny et al. (2018) stated that MSA is not a superset of SOA, pointing out the fundamental differences as MSA with a shared-nothing approach and SOA with the share-everything approach. While the shared-nothing approach is providing a better categorization, it does not differentiate the other aspects of SOA from MSA. Common characteristics between MSA and SOA exist, and both are trying to solve similar business problems.

The service-architecture is a way to address business problems with adaptable software components and systems. Pautasso et al. (2017) argued that MSA is a best-practice approach to SOA as emerging in addressing some of the SOA challenges, such as placing the business logic into the center of the execution transaction, resulting in performance, scalability, maintenance, and operational issues. While SOA tried to simplify the software development work for business problems, the specific technology implementations resulted in significant challenges for organizations to realize the benefits of SOA. Also, Jamshidi et al. (2018) identified the origin of MSA in SOA with the example of the movement in the industry from the simple object access protocol (SOAP) to representational state transfer (REST) because of dissatisfaction with the technical limitations. While SOAP is a protocol with the formal specification, REST is an architectural style providing more freedom in the implementation approach. Having a choice of architectural design and implementation technology is crucial for developers.

The choice of architecture and technology is an evolutionary step for developers and organizations to improve software systems. Gabbrielli, Giallorenzo, Guidi, Mauro, and Montesi (2016) claimed that MSA evolved from SOA with the application of component-based software engineering to a service resulting in MSA. Zimmermann (2016) claimed MSA is an evolution version of SOA as sharing common characteristics that include business focus, multiple architecture concepts, independent technology for implementation, resiliency against failures, no centralized control, and high automation of all related aspects. There is no doubt that the practitioner modified existing architectural concepts to allow them to address business problems efficiently. These new best practices for SOA-based systems worked in their organizations and contexts evolving towards a broader accepted set of adopted practices for MSA-based systems. While researchers will continue to argue over the positioning of MSA in the group of service architecture concepts, multiple concepts influenced practitioners in developing MSA to address their problems.

Domain-Driven Design

Domain-driven design is a prominent approach to establish the boundaries of the business knowledge for the design of MSA-based software systems. Evans (2004) formulated the domain-driven design approach due to developing software for complex business problems in 17 phases to enable alignment to business requirements and flexibility to evolve. The domain-driven design approach simplified the development of code that is closely related to the domain model. Le, Dang, and Nguyen (2018) presented domain-driven design as an approach for software development to capture the domain requirements and implementation approach into a realistic model to enable a direct realization as a code. Containing the complexity of the domain into a manageable chunk is crucial for developers in understanding the requirements for creating a piece of useful software code. While domain-driven design creates a focus around the business problem domain, boundaries are needed for the implementation and operations.

The definition of boundaries is an essential step for solving a problem. Thönes (2015) argued that domain-driven design supports the MSA design by deriving an appropriate size for a problem domain in a bounded context. The appropriate size of the MSA-based system is an essential concern to ensure the team responsible for the MSA can maintain and operate without impacting surrounding services. Also, Zimmermann (2016) asserted that domain-driven design is used to extract the business domain for the definition of MSA using bounded contexts and domain models. Practitioners adopt best practices over time when proven useful and apply to new situations. Similarly, Schwartz (2017) articulated that domain-driven design provides the mechanics to describe the domain aspects by ensuring a high cohesion in a bounded context. High cohesion of business functionality is useful to enable that all relevant software aspects are contained within the control of the developer team. While the domain-driven design enables to create bounded contexts for the development of the MSA-based systems, domain-driven design does not explicitly consider the implementation aspects that the developer needs to create software code.

The implementation aspects for MSA-based systems are of significant concern for the developers and operations teams supporting the MSA environment. Pautasso et al. (2017) presented domain-driven design as one of the tools to design MSA that guides the design and implementation of the code but still did not address all aspects, for example, interfaces or infrastructure components. The technical realization of software code requires decisions on areas that are not related to the domain and a need for the environment for the implementation. Rademacher et al. (2018) listed the challenges of domain-driven design with MSA as of missing the identification of interfaces, endpoints, protocols, and operation parameters that are required for the service implementation. The implementation of the MSA-based system is constraint by the selected software language, the frameworks to support the development, and the infrastructure environment that will host the software. Similarly, Taibi and Lenarduzzi (2018) identified elven problematic architectural practices that impact the software quality aspect of MSA-based systems. The researcher did not focus on particular domain-driven design aspects. However, most of the identified issues can be applied to strategies for domain-driven design. The challenges identified by Rademacher et al. (2018) for interfaces and endpoints are supported by the findings of Taibi and Lenarduzzi (2018). However, the issue of missing the operations parameter for the service implementation is not reported by Taibi and Lenarduzzi (2018). The specific differences in the findings might be the result of the different research focus on domain-driven design and microservice issues by Rademacher et al. (2018) using an example MSA development approach and the research by Taibi and Lenarduzzi (2018) collecting MSA-based bad practices using a survey of practitioners on relevant practitioner events. While domain-driven design supports the development of a design that suits MSA, additional considerations for the infrastructure, interfaces, and operational aspects must be included in the strategies for domain-driven design to enable that the domain-driven design-based model can be implemented as an MSA-based system.

Development and Operations (DevOps)

DevOps deliver MSA-based software code from the development stage into production quickly. Zhu et al. (2016) defined DevOps as the practice in optimizing the time to deploy code commits into the production environment. Laukkarinen et al. (2018) noted the increase of release frequency and 86 percent reduction of time to deploy after the DevOps adoption. According to Erich et al. (2017), the DevOps practice includes people, processes, and tools to support the effort in shorter release cycles. The increased demand from the business pushes the development teams to adopt new ways of working and tools that increase the output of new software releases. The reduction of the time for bringing a software code into production requires an integrated toolchain.

Having an integrated toolchain is crucial for the automation of the MSA lifecycle phases using DevOps practices. Ebert et al. (2016) argued that high-quality software deployment needs automation and integration of the process steps in the software lifecycle. Standardization of tasks allows automation and increases the repeatability of identical output, improving the quality of the software development. Likewise, Laukkarinen et al. (2018) maintained that DevOps requires an automated toolchain to be effective. Typically, in software development, new code moves through multiple stages such as the code is working correctly, the integration with other software systems is intact, the performance is understood, data security is validated, and many other aspects. The efficiency increases when the movement through the development stages is orchestrated by a tool. Also, Callanan and Spillane (2016) noted the importance of creating an automated toolchain for continuous deployment (CD) and automated testing to assure the developers of the correctness of the code. While testing is a need to ensure the correctness of the code, the automation of the deployment activities of the new software code into production is an operational concern. Zhu et al. (2016) asserted that orchestration is needed to manage the pipeline flow for each stage of the software lifecycle. Automation of each stage in the software lifecycle increases the quality and reduces the cycle time to move new functionalities in the production environment for the business users. Establishing integrated tooling comes at the cost of increased complexity; however, the MSA lifecycle management benefits from an automated and integrated toolchain supported by DevOps practices.

DevOps pairs well with MSA as supporting the software development and lifecycle activities. Balalaie et al. (2016) posited a relationship between DevOps and MSA with MSA needing continuous integration (CI) and CD to enable a pipeline for ongoing deployments. The bounded context of the MSA and the supporting tools support a high degree of automating tasks from the lifecycle activities. Similarly, Cerny et al. (2018) asserted that DevOps aligns well with MSA-based applications as supporting the on-demand requirements. The orchestration and automation possibility of the DevOps with MSA increases the agility in addressing business requirements faster. Also, Bass (2018) concluded that the choice of architecture influences the ability to deploy efficiently using MSA as an example. MSA-based systems present characteristics that support automation and orchestration over the complete lifecycle of the software system. Combining MSA and DevOps enables higher efficiency and higher speed in deploying new code into production environments. While DevOps practices and MSA match very well, both concepts require addressing people, processes, and technologies for the adoption.

Continuous integration (CI). CI provides a tool for the MSA adoption in the software engineering process. Ebert et al. (2016) defined CI as a central place to merge, organize, and validate the developer's code commits before building the software package. Automating the effort of merging code parts from several developers and validating the code against a set of test parameters reduces the time to build a new software package. Pallis et al. (2018) argued that MSA-based systems support the frequent code updates to the software systems by using CI practices. The confined construct of MSA-based applications presents fewer challenges for the CI practices to implement as less external parameters impact the automation effort. O'Connor et al. (2017a) asserted that CI benefit from the MSA-based application system in the software engineering process. The software architecture impacts the ability to embrace the CI in the development process fully.

Similarly, Ståhl, Mårtensson, and Bosch (2017) claimed that MSA-based applications support the developability through architectural decisions embedded in the MSA. The bounded context of an MSA-based system keeps the dependencies for the code integration and validation within the software construct of the MSA. The results of the qualitative study conducted by Erich et al. (2017) demonstrated that microservices play a significant role in automating the CI pipeline in the development and operations (DevOps) practice. Tasks executed for code integration and code validation that have fewer external dependencies can be automated with less effort. While researchers argued that a beneficial relationship exists between MSA-based systems and CI, practitioners implemented the CI as a tool in the MSA software engineering process to ease the effort in the code integration and code validation processes.

Migrations to MSA encourage the use of CI as a tool to automate the code integration and validation activities of the DevOps practice. Balalaie et al. (2016) reported for the migration of commercial software to MSA-based application using a CI tool are a crucial initial component to establish the DevOps practice. Using proven processes and tools for a migration allows the developers to focus on the problem instead of diverting effort to repetitive tasks. Similarly, Bucchiarone, Dragoni, Dustdar, Larsen, and Mazzara (2018) presented an experience report of a banking application migration to an MSA application using a fully automated CI pipeline as part of the DevOps practice. Developers benefit from the fast feedback of the quality and success of new code developed when the CI pipeline is automated for migration projects. While the automation of CI pipeline is more straightforward for MSA-based applications, CI is only one component of the DevOps practice and software engineering processes.

Continuous deployment (CD). CD enables deployment of MSA-based application releases into the production environment frequently. Schermann, Cito, Leitner, Zdun, and Gall (2018) defined CD as a practice to orchestrate the activities of testing and deployment of software code before releasing to production. Testing of the new code for applications is an essential set of activities to ensure the usefulness of the application before deploying it into a production environment. Killalea (2016) noted that CD enables a change in responsibilities of the MSA application development process and speed of releasing new software versions faster. The automation of testing shifts the responsibilities of testing input to the developers to ensure the tools have the right use cases for the test automation. The automation of the CD requires to define exit criteria for the testing before the code reaches the production environment.

The standardized quality gates with existing criteria help the developers to get fast feedback on any quality issues and code failures before deployment into production. Zhu et al. (2016) observed that CD allows the developer to release code changes to production independently of the involvement of deployment teams. Standardized parameters for moving from each stage in the DevOps pipeline removes the need of manual effort in code validation and task executions in the deployment as the impact of changes is understood. Callanan and Spillane (2016) reported that developers deployed MSA application code changes using an automated CD pipeline within 21 minutes into the production environment successfully, approximately 200 times faster than before. The confined environment of MSA-based systems and the bounded context of the code supports the automation of the deployment activities using standardized parameters for each quality gate in the CD process. The automation of the CD pipeline reduces the time to deploy code to production significantly with standards needed for automation.

Automating the CD pipeline requires standardization of the testing effort to improve the quality of the code deployments. O'Connor et al. (2017a) discovered that deployment automation did not impact software quality. The fear of losing control by automation is offset by the transparency of the parameters applied to allow the new code to move into the production environment. According to Leppanen et al. (2015), organizations struggled to automate CD pipelines, where security or performance affects the quality of the software code deployed. Security and performance are non-functional requirements that are difficult to validate and therefore challenging to confirm via parameters as part of DevOps toolchain. Likewise, Claps, Berntsson Svensson, and Aurum (2015) ascertained that the fast deployment approach in CD outweighs the impact of any quality issues in software code. The ability to deploy new software code quickly into a production environment supports the possibility of fixing a problem fast as well, therefore limiting the impact to the users. Otherwise, Russo and Ciancarini (2017) concluded that CD improves the quality of code when used in combination with test-driven development (TDD) approach. The automation possibilities of CD allow integrating other frameworks to improve the quality of the code developed. While the researchers did not find that CD enhances the quality of the software, only standardization in the CD pipeline enables fast deployment of code changes into production. To address the quality of the code developed requires integrating other approaches that target the various software quality aspects.

Container Technology

Container technologies simplify the deployment of MSA systems and their management. Jamshidi et al. (2018) argue that containerization and container orchestration influenced the development of MSA through development, deployment, and operations. The use of DevOps practices encourages the use of technologies that support automation and can be made available within short timelines. Fetzer (2016) demonstrates building critical applications fail-stop using MSA applications and secure containers inside an Intel software guard extension (SGX) enclave. The availability of an application is an essential requirement for the usefulness and must be considered for the implementation by the developer. Tarmizi and Shanudin (2017) developed a method for analyzing and designing MSA holistically and using containers technologies for the deployment of the MSA-based system. The selection of infrastructure components must be aligned to the architecture to support the availability, security, and performance aspects of the software systems. Wan, Guan, Wang, Bai, and Choi (2018) evaluated the resource allocation optimization for container services used by the MSA-based system as a deployment platform and establishes the need for a framework to handle the MSA-based system scalability. The modularity of MSA-based systems requires the underlying infrastructure to handle performance requirements appropriate to ensure the usefulness of the application functionalities.

Similarly, Rademacher et al. (2018) addressed the challenges of the domain-driven design approach for MSA using a container execution environment for each bounded-context identified to ensure the communication path, protocols, and message formats are established appropriately. The distributed nature of MSA-based systems must be supported by the infrastructure where containers have an advantage over other hosting platforms. Pallis et al. (2018) presented the Unicorn Framework to address the challenges of the MSA adoption using DevOps practices for monitoring and diagnostic, optimization of auto-scaling, the orchestration of hybrid cloud deployments, and security considerations with the inclusion of container-based technology. The container technology provides the MSA-based systems with the lightweight execution environment for efficient deployment and allows DevOps practices to align with the lifecycle requirement of the MSA-based system.

Relationship of this Study to Previous Research

Research in the adoption of MSA is increasing for technical topics, challenges, drivers, and migration approaches, but none is addressing the required best practices for organizations. While researchers try to understand the pitfall and technological implications, IT product managers need a set of strategies to address the challenges for the agile development using domain-driven design strategies and DevOps practices to deploy and operate the MSA-based system efficiently. Practitioners rely on best practices that have been shown reliably to understand the impact of their organizational environment.

Qualitative studies have investigated the challenges of MSA-based system lifecycle challenges. Zimmermann (2016) examined the MSA-based systems by comparing to SOA-based systems, including nine practitioner questions, to adopt MSA-based systems more successfully. While proposing questions to practitioners allows steering the investigation, best practices will address a broader scope for the adoption of MSA-based systems in organizations. Similarly, Taibi and Lenarduzzi (2018) presented five lessons learned for MSA-based system development for the reported 11 bad practices that impact the MSA-based systems negatively for software qualities attributes related to maintainability, testability, reusability, understandability, and extensibility. Software architecture decisions affect the behavior of the system during the software lifecycle, not only for the user as well as the individuals involved in development and operations. Soldani et al. (2018) analyzed the existing industrial grey literature to identify the technical and operations pains and gains of MSA-based systems. While the classification of pains and gains is providing some input for the development of best practices, it is not sufficient for articulating a comprehensive and proven set of practices for use in organizations. Also, Baškarada et al. (2018) investigated the opportunities and challenges of adopting MSA-based systems by interviewing 19 architects related to MSA development. A detailed description allows the reader to develop a set of practices based, but the research does not present the context for the assessment of the application of the practices. While the researcher established a foundation of challenges and opportunities that exist in the adoption of MSA-based systems, best practices that can be used by individuals in organizations are only partially available. A specific case is a migration from an existing software system architecture to MSA-based systems.

Adopting MSA as part of a migration away from monolithic software systems requires a particular set of practices. Bucchiarone et al. (2018) presented a case study of the Danske Bank's migration to an MSA-based system for the currency conversation system outlining the architecture and implementation approaches for the new system. The description of the design and implementation experience of the transition to an MSA-based system allows us to extract best practices. Still, they do not cover the operational viewpoint for the lifecycle activities. Operationalization is a crucial concern for an organization to maintain the benefits of the lifecycle of the software system.

Efficient development low-risk introduction of new code into production operations (DevOps) and are essential aspects for organizations. Erich et al. (2017) investigated how DevOps is used in six organizations and noted the weak motivation of DevOps to use the MSA-based system. While DevOps does not rely on MSA-based systems, MSA-based benefits from DevOps practices. Similarly, Shahin et al. (2017) presented a comprehensive set of practices for DevOps without considering the architectural systems aspects of the practices. DevOps practices drive automation and orchestration of the software code via a pipeline from the development into the production environment. While DevOps practices can handle many different software architectural constructs, MSA-based required particular treatment to ensure efficient operations of such software systems. Before the development is possible, an MSA-based design is needed to provide the scope of the functionalities and dependencies on other systems are defined.

Domain-driven design is an approach to optimize the design of MSA-based software systems for maintainability and scalability during operations. Tarmizi and Shanudin (2017) presented a holistic approach to analyze and design MSA-based systems by catering to technology, geographic, time, and customer aspects, including container technologies and cloud computing. The need to have best practices available to create boundaries for the development and operations, including the required infrastructure is a concern for an organization to plan and orchestrate necessary resources for the MSA adoption. Also, Rademacher et al. (2018) described the domain-driven design challenges based on a cargo domain model, including the suggestions to overcome the issues. Domain-driven design models cover the business concerns but not the needed infrastructure to the level required for the efficient implementation and operations. While researchers try to identify practical ways of elicitation MSA from the business domain, the best practices need to address the complete technology stack for MSA-based systems. Researchers can't ignore the complexity of the end-to-end process in designing, implementing, operating, and maintaining MSA-based systems.

The gap in the research for strategies of MSA-based systems covering the lifecycle is evident. Researchers have investigated either challenges and motivations for the MSA adoption or investigated parts of the lifecycle related to the MSA adoption only. Strategies covering the design, implementation, operational, and maintenance aspects of MSA-based system adoption to address the end-to-end adoption are not available in the literature. Therefore, I considered this as a gap in the knowledge requesting for an investigation. I used a multiple case study research design to investigate the strategies of domain-driven design and DevOps to reduce the inefficiencies in the MSA system adoption.

Transition and Summary

In Section 1, I presented the introduction to the problem of the inefficiencies in the MSA system adoption and the background of the qualitative multiple-case study. The purpose of this study was to explore strategies for the domain-driven design and DevOps practices used by IT product managers to reduce the inefficiencies in the MSA system adoption. The literature review presented the conceptual framework and the current use in the research of technology adoption. The use of TOE theory is considered enabling an in-depth investigation into the domain-driven design strategies and DevOps practices of an MSA system adoption in an organization. Also, the literature review focused on the MSA adoption, domain-driven design, service architecture, DevOps, and related practices.

Section 2 provides details of the project and justifications for the research method and design of this study. In section 2, the role of the researcher, the participants, the population, and sampling of the participant, including the protocols for data collection and data analysis that has been used for this research, will be described. Additionally, the topic of ethical research, reliability, and validity is presented.

Section 2: The Project

In Section 2, I outline my role as a researcher, provide details of the participants, describe the approach regarding sampling, research method, and design decisions. Also, I discuss the ethical considerations and procedures for data collection, data organization techniques, data analyses. Finally, I examine the reliability and validity of this study.

Purpose Statement

The purpose of this qualitative multiple case study was to explore strategies for the domain-driven design and DevOps practices used by IT product managers to reduce the inefficiencies in the MSA system adoption. The population for this study were IT product managers and DevOps members associated with the adoption and operation of MSA systems at one global consumer goods manufacturer company and one automotive company within Europe. The potential social impact of this study included the possible improvement of the user experience for software systems constructed as MSA used by individuals in their daily life.

Role of the Researcher

The role of the researcher is to conduct a high-quality study. Yin (2014) presented the essential attributes of a researcher for the collection of case study data as the ability to ask the right questions, be a good listener, stay adaptive, have a firm grasp of the issues being studied, and avoid biases. I was the sole researcher who collected the data during interviews and from sources that informed my study. I analyzed the data and developed the report.

The selection of the case can be influenced by the interest in the topic of the researcher (Hyett, Kenny, & Dickson-Swift, 2014). The idea for this study came from experiencing a trend in developing and deploying MSA-based systems as a primary application construct while working as a principal architect for a large international system integrator. While my expertise is in infrastructure systems designs, I had limited exposure to DevOps practices and no knowledge of the domain-driven design of MSA-based systems. Currently, I live in Bavaria, Germany, and work in Austria, Switzerland, and Germany. The proposed case organizations were selected from the consumer goods manufacturer sector and the financial industry in Europe. I had no working relationship with the selected organizations for my study. The participants I chose for the interviewees of my research had no past or current relationship with me.

As a researcher, I conducted the study using the principles and applications of the Belmont Report. The Belmont Report stipulates the ethical principles related to social research as respect for persons, beneficence, and justice, including their applications of informed consent, assessment of risks, and selection of subjects (Friesen, Kearns, Redman, & Caplan, 2017). I ensured that all participants were treated respectfully, equally, and followed the study protocol. I minimized the exposure to harm by presenting a consent form with details of risks and benefits, including the option to not participate in the study at any time.

The mitigation of bias is a crucial aspect of qualitative studies as the researcher is the primary instrument for data collection. A researcher must be aware of potential bias at all phases of the research process to deploy mitigation strategies (Malone, Nicholl, &

Tracey, 2014). Bias can be minimized by combining multiple data sources, many different interviewees, applying a research protocol, and systematically analyzing and presenting the data (De Massis & Kotlar, 2014). I used multiple sources of data that includes interview data, organizational data such as procedure and policies, direct observations, and archival records. I followed the data collection procedures as set out in the data collection section.

Interview data is the leading source of evidence in case studies as it provides the researcher with rich and detailed data of the case. Using an interview protocol allows the researcher to increase the quality of the data collected and to strengthen the reliability of the study (Castillo-Montoya, 2016). The rigor in the development of a semistructured interview guide improves the objectivity and trustworthiness of research (Kallio, Pietilä, Johnson, & Kangasniemi, 2016). I used an interview protocol to ensure that each participant was treated equally and to ask the same set of questions. I recorded the interviews and took notes so that accurate data was gathered.

Participants

The participants for my research were practitioners of domain-driven design strategies and used DevOps practices for MSA adoption in two case organizations based in Europe. Robinson (2014) asserted that qualitative interview-based studies increase the validity of the study when selecting a systematic sampling approach to source participants. Similarly, Gentles, Charles, and Ploeg (2015) recognized the importance of criteria in selecting knowledgeable participants for the quality of the data collection. Moser and Korstjens (2018) suggested recruiting participants who have an in-depth understanding of the phenomenon and willing to share their knowledge. I applied the eligibility criteria that were aligned with my research question and the phenomenon of my study. For this study, I used participants that met my criteria to be able to collect high-quality data. Eligible participants have been individuals with an overall practical experience of domain-driven design strategies, DevOps practices, and MSA adoption for at least 2 years and a minimum of 1 year working in the case organization on MSA-based systems.

To gain access to the sources of knowledge relevant to answer the study questions in the case organization, I required help for the identification of individuals based on criteria. Robinson (2014) suggested using an individual in the case organization, a gatekeeper, to support the identification and encouragement of the participants for the study. Dempsey, Dowling, Larkin, and Murphy, (2016) argued that gatekeepers are more supportive when the researcher shared relevant study details, potential benefits of the participation, and is available for clarifications. Moreover, Riese (2019) added that the researcher should choose the gatekeepers carefully. This is because gatekeepers influence and structure the interactions between the participants and the researchers. I used my LinkedIn network to identify any possible gatekeepers within the case organization. To enable the gatekeepers as a promoter for my study, I described my research, including potential benefits for participation. Once the permission for my research was granted by the IRB, I worked with the gatekeepers in both organizations to identify the individuals based on my criteria for participation. The gatekeepers provided me with the name, email address, phone number, and availability of potential participants.

Establishing a rapport with my participants allowed me to collect data efficiently for my research. Dempsey et al. (2016) recognized that establishing a rapport and building a trusting relationship is essential for interviews with sensitive topics. Similarly, Garbarski, Schaeffer, and Dykema (2016) concluded that the responsiveness and empathy of the researcher to the interviewee's answers determine the level of rapport and quality of data collected. Also, Jenner and Myers (2019) reported similar findings for establishing a rapport for face-to-face interviews compared to Skype-based interviews. Consequently, Vasquez-Tokos (2017) recognized the advancement of building a rapport when the researcher has a shared experience and is an empathic listener for gathering in-depth data. Hence, Arsel (2017) suggested to clarify the interview procedure, roles, expectations from the participant, explain the study, the researcher's background, and motivation of the research for this study. I established a rapport with the participants by acting transparently and ensured the autonomy of the participant at any time during the research. I collaborated with each participant in selecting the interview setting for comfort and privacy. I shared my biography with the participant before the interview and elaborated on my background and work experience regarding domain-driven design strategies, DevOps practices, and MSA to establish a common ground for the interview.

Additionally, I emphasized that the interview is to understand their experience on the topic of my research. Furthermore, I clarified the participant's right to refuse to answer any questions and to cancel the interview at any time. During the interview, I listened carefully to identify opportunities for insightful next questions at the right time, showing an interest in the participant's experience.

Research Method and Design

A researcher must select an appropriate research method to answer the research question. The research methods, such as the qualitative method, the quantitative method, and the mixed-method, allow the researcher to investigate the phenomenon using a specific plan and procedure as the research design. I chose the research method and research design based on the research questions and data availability for answering the research questions.

Method

I chose the qualitative research method to investigate the domain-driven design and DevOps strategies that IT managers apply in their environment for MSA adoption. Bansal, Smith, and Vaara (2018) argued that investigating a complex phenomenon using qualitative methods allows the researcher to discover new insights and enables new viewpoints. Leppink (2017) framed qualitative research as focusing on answering the research questions by collecting rich qualitative data, including text, words, visual and other artifacts. Moser and Korstjens (2017) described a research approach as enabling an in-depth understanding of the real-world phenomenon. Also, Quick and Hall (2015a) indicated that the qualitative method allows the researcher to increase the understanding of the problem by investigation of experiences, behaviors, and perspectives of the involved individuals. Taguchi (2018) argued that qualitative methods enable the researcher to investigate a more significant number of aspects and represent qualitative data as evidence for the finding. The qualitative approach helped me to investigate the complex concepts of domain-driven design, DevOps strategies, and MSA adoption deeply by collecting data through open-ended interview questions and artifacts from the participants.

I did not select the quantitative method for my research goal as many attributes, and the research question would not be addressed by this method. Quantitative research allows the testing of a hypothesis using a broader set of data from a more generalized sample of participants (McCusker & Gunaydin, 2015). The design of this study is not testing a hypothesis that would warrant a quantitative method. The data collected by quantitative research is through an instrument or experiment and subject to statistical analysis (Boeren, 2018). I collected data using interviews, observations, organizational documents, and field notes without manipulation of the participant. The data was analyzed by identifying categories and themes to derive insights, without the need for statistical methods such required in quantitative research. The quantitative research seeks to understand the phenomenon by measuring variables and the identification of the relationship between these variables (Taguchi, 2018). This study did not measure variables of strategies for the domain-driven design and DevOps practice to identify a relationship to the inefficiencies in the MSA adoption. Therefore, a quantitative method was not justified for my study.

While I initially considered the mixed-method research which is a combination of the qualitative and quantitative method, to be suitable for my study, the research question could be answered by the qualitative method solely. Mixed-method research supports the exploration of complex phenomena by enhancing the quantitative data with in-depth qualitative knowledge to increase the power of the findings (McCusker & Gunaydin, 2015). The combination of the qualitative and quantitative methods requires the researcher to address the validity of each method in the mixed-method research to ensure overall validity (Leppink, 2017). With a mixed-method the researcher needs to integrate the qualitative and quantitative data in such a manner to reinforce the answer to the research questions to ensure the support of the findings (Molina-Azorin, 2016; Taguchi, 2018). The number of participants available for my study with knowledge on the topic of strategies for domain-driven design and DevOps practices including MSA adoption experience was limited. Hence, the quantitative data approach would not establish a significant statistical power to strengthen the inference from the data collected. Therefore, I considered the quantitative method and mixed-method as not suitable for my research.

Research Design

I used a multiple case study design for this research. A multiple case study design allows to investigate multiple cases and identify similarities and differences between the cases (Yin, 2014). Similarly, Vohra (2014) described that a multiple case study design allows applying a literal replication to confirm or disconfirm the findings. The researcher can use a multi-case study to gain an in-depth insight into the case phenomena and possible shared characteristics by using a multi-faceted enquire with various types of data sources (Carolan, Forbat, & Smith, 2016). Also, Hyett et al. (2014) described a case study as an investigation into the complexity of a phenomenon using multiple sources for analysis. Case studies have been the dominant research design to investigate technology strategies in organizations (Dasgupta, 2015). For this study, a case study was suitable because I researched the phenomena of MSA adoption and the related strategies of domain-driven design and DevOps that have unclear boundaries within the case organizations. Using a multiple case study design enabled me to investigate in detail the strategies for domain-driven design and DevOps practices applied to MSA in two organizations operating in different industries. I used a multiple case study to be able to replicate the procedure for the investigation to the two case organizations to strengthen the results of the findings.

In a case study research, reaching the data saturation is crucial to ensure the high quality of the research. Data saturation is reached when the information collected justifies the findings and conclusions (Constantinou, Georgiou, & Perdikogianni, 2017). Fusch and Ness (2015) explained data saturation as a concept when no new data is emerging in the form of codes or themes, allowing the researcher to replicate the study. Moreover, Lowe, Norris, Farris, and Babbage (2018) suggested measuring the saturation based on the number of available themes and the average number of themes in the observation to determine the level of saturation reached. Hagaman and Wutich (2017) reported that the number of interviews and divergence of the participants influences the number of interviews needed to reach themes saturation. I evaluated the level of new themes emerging from the interview data collected to identify if I reached saturation in my study. I adjusted the number of participants to ensure data saturation.

As a researcher, the design of the study should demonstrate trustworthiness. A high-quality case study research requires to collect and converge relevant data through documents, interviews, archival records, direct observations, and physical artifacts to improve the power of the findings (Yin, 2014). Korstjens and Moser (2018) suggested using triangulation with multiple sources of data establishing thick and rich qualitative data as a basis for the analysis-enhancing credibility. Smith (2018) argued that in a case study using multiple sources of data and data triangulation enhances the trustworthiness of the research. Similarly, Baškarada (2014) emphasized to collect data from multiple sources to allow for data triangulation and convergence of the findings as a strategy to establish rigor in the research. I collected rich and thick data descriptions using semistructured interviews and organizational documents relevant to the strategies for domain-driven design and DevOps practices in the context of MSA adoption from two different organizations. I used data triangulation and methodical triangulation for the data collection to enhance the trustworthiness of my study.

Confirming the interpretations of the collected data is enhancing the credibility of the research. Member checking allows reducing bias from the researcher by actively validating the interpretations of the data with the participant (Birt, Scott, Cavers, Campbell, & Walter, 2016). Hays, Wood, Dahl, and Kirk-Jenkins (2016) asserted to use ongoing member checking for ensuring the accuracy of the interpretations of the interview data collected to enhance the credibility. While Hadi and Closs (2016) suggested member checking as a strategy for validation of the conclusions from the interview as the most critical method, the interpretations should not include synthesized data from some participants. Therefore, I conducted member checking with the participants to validate the accuracy of my interpretations of the participant’s collected data only.

Besides the case study design, I considered other designs for qualitative studies such as ethnography, phenomenology, and narrative for my research. Ethnographic research allowed to investigate everyday life in a group under study (Billo & Mountz, 2016). Grossoehme (2014) suggested using an ethnography design for investigations of the impact of a phenomenon on a group of people. Likewise, Quick and Hall (2015b) presented ethnography as an exploration of a cultural phenomenon with the researcher embedded into the culture. The researcher immerses himself into the group to study the culture over a certain time to gather relevant data for describing the culture (Draper, 2015; Trnka, 2017). Interviews, observations of the participants, field notes, and archival records enable the researcher to reveal patterns to gain insights (Billo & Mountz, 2016; Kruth, 2014). The focus of my study was to investigate strategies the IT manager applied and not to study a cultural group in the organization. Additionally, the study of strategies did not require to observe the participants over an extended period. Therefore, I did not select an ethnographic design for my research.

The phenomenological research intends to understand how individuals experience the phenomena that are investigated (Pietkiewicz & Smith, 2014). Matua (2015) recognized phenomenological as suitable to provide a detailed account of the phenomenon in structuring the essence of the experiences from the individual. Also, Wilson (2014) argued that the power of phenomenology researchers to elicit insights into the experiences of individuals were impacted by the phenomenon. The researcher focuses on specific situations that individuals perceived and express the essence as a written account of the phenomena (Robertson & Thomson, 2014). Lingis (2017) asserted that phenomenological research allows expressing the individual experience with the contextual details for the readers to understand the impact of the phenomenon. Chan, Walker, and Gleaves (2015) justified the use of phenomenological research design as being able to describe the lived experience of using smartphones as a learning platform by students by the researcher through language. (Chan et al., 2015; Gill, 2014; Skea, 2016). My study was not focused on exploring and describing the lived experience of IT product managers. Similarly, I did not investigate how the strategies for domain-driven design and DevOps are perceived by individuals in the organization, instead, exploring the experiences of the MSA adoption. Therefore, the phenomenological design was not suitable for my research.

Narrative research explores the experience of individuals (Lilgendahl et al., 2018; Rosiek & Snyder, 2018; Yamagata-Lynch et al., 2017) A narrative design allows to present the events and situations as a story that is experienced by the participants (Thompson Long & Hall, 2018; Yamagata-Lynch et al., 2017; Zurlo & Cautela, 2014). A story creates a structure that can be shared and enhanced with explanations by the researcher of the lived experience during the phenomena (Bruce, Beuthin, Sheilds, Molzahn, & Schick-Makaroff, 2016; Carmel-Gilfilen & Portillo, 2016; Grysman & Mansfield, 2017). The focus of my research was not to describe the life of the IT manager but the strategies they use. Therefore, the narrative design was not supporting my research goal.

Population and Sampling

The population for my multiple-case study consisted of IT product managers for MSA-based applications from the case organizations within a global consumer goods manufacturer company and a financial company with headquarters in Europe. Palinkas et al. (2015) argued that purposeful sampling strategies support the selection of sources with in-depth knowledge of the phenomenon using criteria for inclusions and exclusions of relevant individuals. Likewise, Benoot, Hannes, and Bilsen (2016) described that purposeful sampling allows the researcher to include information-rich participants of the topics for an in-depth examination of the study problem. Robinson (2014) suggested using purposive sampling to ensure the inclusion of participants in the final sample that can provide rich and in-depth insights into the phenomenon. I used the purposeful sampling approach to select the participants for the data collection interviews. The population of my study included IT product managers and DevOps members with experience of MSA, DevOps, and domain-driven design for at least two years and working on MSA-based systems and of one year in the case organization with headquarters in Europe that had not any past or current working relationship with me. The case organizations had at least 5 MSA-based systems in operation for at least two years to ensure that a sufficient population of IT product managers and DevOps members related to the MSA-based systems is available for my investigation. I did anticipate a population of approximately 30 possible participants per case organization using the defined legibility criteria for the participants. Therefore, I approached the entire population as potential interview participants of my research.

Reaching data saturation was essential for my research to establish a high-quality study and the power of the findings. Constantinou et al. (2017) reported that data saturation is reached by the eight's qualitative interview after rearranging the interviews for the evaluation of data saturation. For the researcher, it is crucial to identify that data saturation is achieved after all, regardless of the threshold for the interviews. Similarly, Hagaman and Wutich (2017) identified a range of 12 to 16 interviews for a focused topic and homogenous group of population to reach data saturation. Having a homogenous group for the interviews is reducing the need for a more significant number of interviews to reach data saturation. Also, Gentles et al. (2015) presented that researchers choose four to ten participants for multiple case studies as a sampling size to reach data saturation by applying selection criteria. The application of selection criteria requires selecting case organizations with a sufficiently large population to enable enough interviews to reach data saturation. While Constantinou et al. (2017) identified the data saturation mark around eight interviews, Hagaman and Wutich (2017) suggested 12 to 16 interviews and Gentles et al. (2015) presented something between four to ten as a range of interviews; there is no clear guidance to identify a number of interviews to reach the quality criteria of data saturation. However, collecting data from approximately eight participants per case organization and continuously evaluating the data saturation would allow me to adjust the sampling size by adding one or more participants in case of not reaching data saturation for the case organization. Having a sufficient large sample size of participants at each case organization for the interviews available was crucial to reach data saturation. I selected large organizations to reach the anticipated population of 30 participants per case organizations to allow for the flexible increase of the sampling size if required. Therefore, I targeted a sampling size of approximately eight participants per case organization to reach a total sample size of approximately 16 participants. However, I increased the sample size to 9, to confirm that I reached data saturation at each case organization. The purposeful sampling criteria were the years of experience in the topic of domain-driven design strategies, DevOps practices, and MSA adoption. I selected participants with most years of experience as my sample for the interviews from each case organization. While the size of the sample for the interviews plays a role in data saturation, the quality of the data source is another aspect that influences the level of data saturation.

The use of data sourcing with immense knowledge on the research topic, including additional documentation, provides the basis for rich data and thick data collection. Constantinou et al. (2017) presented data saturation as evidence of enough information when rich and thick data is collected to address the research question. Gentles et al. (2015) described saturation of reaching the level of redundant information of the data collected, leading to no new insights. Malterud, Siersma, and Guassora (2016) asserted saturation as no further information is present in the data that adds to the development of the theory. Data saturation can be reached by collecting rich and thick data from the sources until no new information can be extracted. My sampling criteria focused on the most knowledgeable participants of the case organization. Therefore, I reached data saturation by collecting data from interviews and organizational documents related to DevOps, domain-driven design, and MSA adoption strategies, methodologies, best practices, and processes to enable a rich and thick data set. While data saturation is one approach to ensure high-quality research outcomes, additional methods and approaches should be incorporated to increase rigor and validity.

Triangulation is an approach supporting the quality and validity of the research. Abdalla, Oliveira, Azevedo, and Gonzalez (2018) described methodical triangulation to collect the data using multiple methods. Drouin, Stewart, and Van Gorder (2015) noted the different focus on the data by using different approaches to collect the data to enable more significant insights from the data. Joslin and Müller (2016) defined the within-method triangulation for different data collection approaches used for the same research design. Methodological triangulation uses multiple methods to collect and analyze data to support a complete understanding of the phenomenon. Member checking enables the participant to revise and enhance the researcher’s narrative until it is correct and complete, which supports the objective of data saturation. Therefore, I prepared a summary of the interview detailing my understanding for each participant to validate as part of the member checking. Additionally, I facilitated the methodological triangulation by establishing a research database and I used a transcription service to enable text data for the recorded interviews. Furthermore, I used multiple analysis methods for the analysis of the data which included open coding, axial coding and categorization of the data for the identification of themes.

Interviews of IT product managers are the primary source for collecting the qualitative data for my multiple case study. The location and timing of the face-to-face interview can impact the quality of the data collected (Dempsey et al., 2016). Ranney et al. (2015) argued that the location and site environment influence the quality of the data collected. Likewise, Gagnon, Jacob, and McCabe (2015) suggested that the researcher should consider the various location constraints and the possible information collected in selecting a suitable place for the interview. Therefore, I collaborated with the participants to identify the most suitable time, location, privacy, and medium used for the interview. While the location of the interview is essential for face-to-face interviews, phone calls and remote video sessions enable additional choices for the interview location.

Skype is a choice for the researcher to create a suitable research environment and interview location. Jenner and Myers (2019) reported no difference in private interviews between Skype and in-person interviews. Similarly, Oates (2015) concluded that Skype is a viable option for semistructured interviews with more choices to select a suitable and safe location for the participant. Seitz (2016) maintained that the researcher could create a viable research environment for the participant via Skype by the participant selecting location and time. The use of Skype increases the options of locations and comfort for the researcher and participant to conduct the interview. Hence, I offered Skype sessions to enable the participant to choose the most convenient time slot and location setting for the interview. The interviews lasted for about 60 minutes, and I followed the interview protocol steps to gather relevant and rich and thick data for my research.

Ethical Research

I conducted my study in line with the ethical research standard of Walden University to protect the participants from harm. Øye, Sørensen, and Glasdam (2016a) concluded that ethical challenges might appear in any phase of qualitative research.

Consequently, Borrett, Sampson, and Cavoukian (2017) proposed to proactively apply ethical consideration continuously during the research to protect the participant from harm. The Belmont Report described three basic ethical principles of respect for persons, beneficence, and justice for research involving the human subject (United States, 1979). Therefore, I ensured that all my activities, processes, and methods related to this research were ethical and protected the participants of my study from harm. The ethical principles and guidelines of the Belmont Report guided my research. I also completed the NIH training course for researching while protecting human subjects (see Appendix A).

The institutional review board (IRB) evaluates the level of protection of the research subject and the value of research contribution before approving my study. Morris and Morris (2016) maintained the challenge of the IRB in determining that the expected benefits of the research over-compensate the risk impact on the participant of the study. I requested a signed letter of cooperation from an authorized leader of each case company. Before collecting the data or conducting interviews, I obtained approval from Walden University's IRB for my study with the approval number 08-28-19-0630728.

The base for my ethical research is the informed consent and voluntariness of the participant. Mamotte and Wassenaar (2015) asserted obtaining voluntary informed consent from the participant of study as the cornerstone of ethical research. Also, Kadam (2017) indicated that the quality of the information and the opportunity of the participant to ask questions increases the understanding of the research. Consequently, Biggs and Marchesi (2015) argued to limit the number of words and maintain readability. Therefore, the consent form and information of my study were limited to fewer than 1250 words, including my availability for clarifications. Doody and Noonan (2016) recommended that the researcher should include a set of essential components in the consent form to support the decision of the potential participant. The content of the consent form included the intent of the study, risks, confidentially, privacy, benefits, and the right to withdraw anytime without consequence.

I distributed the consent form to the potential participants over email that the gatekeepers of the case organizations identified based on criteria for my research. Mason and Ide (2014) noted the reduction of coercion between researcher and participant using email and the increased of control of the participant to answer questions in their sequence from the researcher. Only once the participant replied with “I consent” via email, I contacted the participant for scheduling of an interview session. Quick and Hall (2015b) emphasized to elaborate the right to withdraw to help the decision making of the participant. Also, Foe and Larson (2016) asserted as a best practice provides the possibility for participants to interact with the researcher for a better understanding of the research impact. During the collaboration for scheduling the interview, I clarified any questions from the participant regarding the research and elaborated the right to withdraw at any point in time. Gagnon et al. (2015) recognized the distress of participants in answering questions at public and private locations during the interview. Therefore, each participant was able to choose a suitable schedule and select a convenient place for the Skype-based interview. McDermid, Peters, Jackson, and Daly (2014) recommended providing the right to withdraw at the beginning of the interview, including an agreement for recording and after the data collection session for publication of information. Before the recording of the interview or meeting started with the participant, I asked if the participant wants to withdraw from the study to ensure their autonomy was intact for participation. The participant would be able to withdraw anytime from the interview and participation in this study without naming a reason. I would destroy all collected data from the withdrawing participant, and I would not use the data in my research. Robinson (2014) argued that incentives for participation in an interview might create distress and support made-up data to get out of the situation. Therefore, I focused on the shared value of the research for the participants instead of incentives to avoid stress for the participant.

Assuring the privacy and confidentiality of all data that I collected were essential components to safeguard the participant from harm. Saunders, Kitzinger, and Kitzinger (2015) asserted that confidentiality through masking of the data requires to include the context to ensure no traceability of the identity of the participant. Similarly, Petrova, Dewing, and Camilleri (2016) reported using codes for anonymization of the participant's identity, including removal of gender and uniquely human features. To protect the privacy and confidentiality of the participants, I masked each participant's name with a unique code that consistent of a case id and a sequential number, for example, AP01 for Participant #1 of Case Organization A; BP01 for Participant #1 of Case Organization B. Places were replaced with generalized descriptions, for instance, Place01 or Country01. Similarly, contextual participant sensitive information was masked as well; for example, job descriptions will be replaced with Job01. The mapping of each participant code to names, email addresses, phone numbers, and all other codes will be maintained in an encrypted text file. Doody and Noonan (2016) recommended protecting all electronic data with a password and stored in a locked facility to maintain confidentiality. All my research data was stored in my encrypted computer, and any physical artifacts were securely locked in a safe. The Institutional Review Board (IRB), doctoral research committee, and I have the right to access. The electronic audio recordings of the interviews were securely destroyed at Rev.com, Inc., once the transcription and validation of the interviews were completed. Starting with the publication date of my study, I will retain all research data on an encrypted pen drive in lockable storage for five years. After five years, I will securely delete all data of this research on the pen drive.

Data Collection

Instruments

For this qualitative multiple case study, I was the sole researcher and primary collection instrument. In qualitative research, the researcher is the primary instrument for collecting the data for a high-quality case study as flexible to adapt to new situations during the interactions (Yin, 2014). Interviews were the primary source of data for my research. Multiple sources of data are typically collected by the researcher to understand a complex phenomenon (Yazan, 2015). Additionally, I requested that the participants bring any material such as organizational documents, procedures, and multimedia files that are relevant for strategies in domain-driven design, DevOps for MSA adoption to the interview.

Developing a protocol for a semistructured interview contributes to the trustworthiness of the study (Kallio et al., 2016). I used a protocol as a guideline for conducting the semistructured interviews to collect the data from the participants. An interview protocol guide supports the researcher to gather rich data in a structured manner and still allow for adjusting to the information received during the interview (Castillo-Montoya, 2016). The protocol covered the steps of the interview process including (a) the purpose of the study, (b) the reason for participation, (c) the risks and benefits of participation, (d) the ethical approach, (e) the confidentiality, (f) the interview questions including follow-up questions, and (g) conclusion of the interview (see Appendix C).

The researcher needs to evaluate what the participant can reveal based on the interview questions for the research question of the study (Arsel, 2017). The interview questions were open-ended and based on the research question of this study. The interviews were recorded after the participant's consent and used for the transcription afterwards. Field notes support the researcher in commenting useful details as part of the interview progress and documenting the possible rationale for the adoption of the protocol (Teixeira Vinci, Lopes Rijo, de Azevedo Marques, & Alves, 2017). During the interviews, I took field notes and recorded any observations of the environment, events in the interview process, about the participant, and identified keywords using the protocol in Appendix C.

I used member checking to improve the reliability and validity of my study by following up with each participant to review my interpretation until no new information emerges. Birt et al. (2016) described member checking for the interview as a possibility for the researcher to verify and confirm the interview transcript. Chase (2017) defines member checking as a methodical approach to confirm that the participants' view was captured accurately. Developing an integrated process for member checking enhances the trustworthiness of the results when executed with rigor. Therefore, I employed data triangulation using the data from the interview and organizational documents as well as methodological triangulation using coding, research database, and interview transcription to maximize the reliability and validity of my study. Scrutinizing the protocol with other researchers or conducting appropriate pilot tests for the feasibility of the data collection increases the trustworthiness of qualitative research (Kallio et al., 2016). To enhance the trustworthiness of my research, I practiced the execution of the protocol and the questions in advance to identify the possible need for adjustments before the implementation.

Data Collection Technique

For this study, I used Skype-based interviews with a semistructured approach, member checking via email and Skype, and document analysis as primary methods for data collection. The data collection started after I received the approval from Walden University's Institutional Review Board (IRB) for my research. For both organizations I used in my multiple case study, I obtained a letter of cooperation.

Recordings of the interviews supports the researcher to increase the accuracy and quality of the research. Nordstrom (2015) described the use of audio recording as an advantage to the reduction of bias and allowed the researcher a more human-oriented interview. Also, Butler (2015) proposed to include field notes of the interview to enhance the transcription for the analysis and improve the rigor of the research. While the recording of an interview is advised either as a primary or backup data source, Reynolds, Choi and Lee (2018) suggest using two recording devices to avoid loss of audio records that would impact the research quality. Each interview was audio-recorded using a dedicated voice recorder on my smartphone and the call recorder application for Skype on my personal computer. Both audio recording devices used encryption of the internal data storage to protect the information stored. As part of the meeting, I took notes during the interview, recording essential events and keywords. The interview protocol supported me in keeping the interview on the topic. In case a participant would bring any relevant organizational documents to the interview, I would request for soft copies to be emailed to me. Additionally, I collaborated with the gatekeepers to obtain any relevant documentation related to domain-driven design strategies, DevOps practices, and MSA adoption approaches used in the organization.

Member checking is enhancing the reliability and validity of the data collection. Iivari (2018) asserted that member checking allows the researcher to increase the validity by receiving confirmation for interpretation of the data collected. Chase (2017) emphasized member checking to increase the knowledge of the research. Additionally, Naidu and Prose (2018) asserted that the researchers should include member checking in a controlled manner with the participants. Consequently, Birt et al. (2016) proposed a structured process for synthesized member checking to minimize the stress to participants and to report the outcome for enhancing the trust of the results. At the end of each interview, a member checking approach was agreed with the participant and conducted over email or Skype. I emailed a synopsis of the interview and the transcribed interviews in advance to the participants before scheduling a session. I provided each participant with the opportunity to confirm and correct the information that I have documented, and if needed I might ask additional questions relevant to improve the accuracy of the information to reach data saturation. For each participant who provided new information, I repeated the member checking approach until no further information was emerging.

For the transcription of each interview and member checking session, I sent the audio recording securely to a third-party transcription service named Rev.com, Inc. Rev.com, Inc. signed the non-disclosure agreement to safeguard the end-to-end protection of participants privacy. To ensure the correct transcription of the audio recordings, I used NVivo to validate the transcription word by word for accuracy. Each version of the transcribed interviews, transcribed member checking session, and organizational documents were part of the research data.

The use of the data collection technique had advantages and disadvantages for my study. Alshenqeeti (2014) described the benefits for the researcher interviewing a data collection method to elicit new information in a natural way of conversation. The choice of using semistructured interviews enabled me to keep on the topic with the participant during the interview. Also, Hammersley (2017) observed the impact of the questions asked to the participant related to the quality of information they provided as a response. The interview questions were open-ended and related to the experience of the participant with domain-driven design strategies, DevOps practices, and MSA adoption. Also, Ranney et al. (2015) asserted that travel requirements of the participant and environmental conditions such as the location, the noise level, the room temperature, and other factors might impact the quality of the information provided. Deakin and Wakefield

(2014) noted that Skype interviews allowed participants' and researchers' flexibility in selecting suitable locations and time slots. I offered multiple options to the participant to choose a convenient time to take the interview and the choice of location for Skype-based sessions at their workplace, at home, or any other place.

Data Organization Techniques

In qualitative research, the organization of the collected data is essential for the researcher to discover insights from the data set. Zamawe (2015) asserted that a computer-aided qualitative data analysis software (CAQDAS) used for data management is crucial for the researcher to analyze the data sets. Similarly, Maher, Hadfield, Hutchings, and de Eyto (2018) noted that for the rigor of the qualitative research, CAQDAS aides the data management and retrieval functionality for the data analysis. Also, Cope (2014) concluded that using CAQDAS supports the researcher in data management to immerse in the data for the analysis. I used NVivo 12, a CAQDAS, as a research database to catalog all data collected in my study. All my research data, whether in electronic format, including audio recordings, notes, journals or as transcribed interviews was stored in my encrypted storage using Boxcryptor and FileVault.

Categorizations and labels enable a structure for the researcher to immerse into the data. Castleberry and Nolen (2018) described the classification as the process of the disassembling with CAQDAS facilitating the organization of the codes. Also, Ose (2016) concluded that an efficient approach for the structure and sorting of the codes enabled the researcher to focus more on the data analysis. Moreover, Robins and Eisen (2017) described the advantage of using CAQDAS codes with query functionality in a large-scale qualitative study. Therefore, I used NVivo for the organization of codes and categories of my research during the analysis of the collected data. To protect the privacy and confidentiality of the participants, I anonymized each participant's name with a unique code that consists of a case identifier and a sequential number, for example, AP01 for Participant #1 of Case Organization A; BP01 for Participant #1 of Case Organization B. The mapping of each participant's code to name, also organization code to the organization was maintained in an encrypted text file. Additionally, I tagged each data source in the research database and labeled each artifact to establish a chain of evidence.

Keeping a reflective journal supports a researcher to change the view of the research. Vicary, Young, and Hicks (2017) argued that journaling within the same CAQDAS holding the collected data enhances the researches' capabilities in the analysis of the data improving the quality of the study. Also, Blair and Deacon (2015) concluded that a reflective approach during fieldwork might support the researcher's decision to adjust methods and activities during the data collection. Likewise, Ortlipp (2008) emphasized to use a reflective research journal to record the changes made by the researcher transparently. I kept a reflective journal in NVivo, recording my experiences, observations, events, decisions, and methods during the research study. I used the notes periodically to enhance my research knowledge evaluating my selection of methods and assumptions in the research progress.

The research database held all data that I collected, including recorded audio files of the interviews and members sessions, transcriptions, organizational documents in electronic form, field notes, and my reflective journal. Starting with the publication date of my study, I will retain all research data on an encrypted pen drive in lockable storage for five years.

Data Analysis Technique

The data analysis for my multiple case studies required an overall analytic strategy to gain insights from the data sets. Yin (2014) suggested that the researcher should develop a strategy for analyzing case study data by using one of the four general strategies such as theoretical propositions, ground up, case description, rival explanations. Also, Watkins (2017) indicated to use a comprehensive strategy for analyzing qualitative data. Drouin et al. (2015) recognized the inclusion of an analysis strategy as a way to improve the level of insight into qualitative research data. Abdalla et al. (2018) concluded to develop multi-method strategies for investigating qualitative material to enhance the credibility of the findings. I used multiple methods for investigating the data to increase the rigorousness of my research.

The strategy was a triangulation of multiple data sets and multiple methods for the data analysis of this study. Renz, Carrington, and Badger (2018) explained four types of triangulation, namely methodological triangulation as the use of multiple methods, the investigator triangulation as the use of more than one researcher, theoretical triangulation as the use of multiple theories, and data triangulation as the use of multiple analytical approaches to increase the validity of the study. Also, Hadi and Closs (2016) noted triangulation as a method to ascertain the validity and credibility of the findings from multiple data sources collected with multiple methods. In contrast, Varpio, Ajjawi, Monrouxe, O'Brien, and Rees (2017) argued the importance of the researcher to not only focus on the sources for triangulation but also on the methods of analysis to maximize trustworthiness of the research. For this study, I used triangulation to increase viewpoints for the analysis by using multiple data sources and methods for collection and analysis.

The choice of triangulation strategy was essential to enhance the creditability and trustworthiness of my study. Joslin and Müller (2016) described data triangulation as the use of multiple data sources, times of data collection, the location of data collection, and persons gathered the information. Similarly, Jentoft and Olsen (2019) demonstrated the use of data triangulation to increase the depth of insights into the data collected from interviews. Consequently, Smith (2018) proposed to use data triangulation in case studies to increase the contextual understanding and creditability of the research. I collected data from multiple organizations, multiple participants by using interviews, including obtaining organizational documents in a short period. Therefore, I applied data triangulation using interview data and organizational data for my study to enhance data and increase the level of understanding for the analysis.

Methodological triangulation increased the quality of the findings in my study. Drouin et al. (2015) indicated that methodological triangulation using multiple data collection methods, multiple data sources, and many analyzing methods increase the depth of insights by evaluating different aspects of the data sets. Accordingly, Renz et al. (2018) depicted methodological triangulation of at least two data collection design approaches, one as within-method triangulation and the other as across-method triangulation. Similarly, Jespersen and Wallace (2017) demonstrated to enhance data insights using methods triangulation to mitigate the weakness of each single data collection method. In addition to the data triangulation, I used the within-method of methodological triangulation and analyses for the data collected from semistructured interviews, field notes, and organizational documents relevant to strategies of domain-driven design, DevOps practices, and MSA system adoption to increase the creditability of my study.

Following a process collected data for the analysis of the improved the creditability of my study. Jamieson (2016) described the process of analytical analysis in five steps starting with data preparation, immersion in data, coding, generating themes, and synthesis. Similarly, Percy, Kostere, and Kostere (2015) defined a generic analysis process for the analysis of qualitative data that can be adjusted to qualitative research approaches. Also, Moser and Korstjens (2018) recognized that the researcher would move back and forth between the process steps to identify valuable insights. I used a five-step process of data preparation and organization, immersion in data with reviewing of the transcripts, identification of codes including categorizations, followed by the identification of themes related to strategies of domain-driven design, DevOps practices for MSA adoption, and as a final step, synthesizing the findings. While the process provides the structure for the analysis, the method of analysis is depended on the expected research outcome.

The approach to qualitative analysis was influenced by the purpose of my research. Crowe, Inder, and Porter (2015) depicted two qualitative analysis methods, namely of thematic analysis as a process to identify patterns from the data and the content analysis identifying the frequencies of categories emerging in the data. Jamieson (2016) presents the thematic analysis as the most applied analysis method for studies that target a deeper understanding of the phenomenon. Likewise, Percy et al. (2015) asserted that thematic analysis allows interpreting data in case studies that collected qualitative data from interviews. Clarke and Braun (2017) describe the thematic analysis as a method to identify and interpret critical data points relevant to the research question.

Consequently, Nowell, Norris, White, and Moules (2017) proposed a six-phase process for conducting a thematic analysis to interpret textual data collected in qualitative research. However, Castleberry and Nolen (2018) suggested using computer-assisted qualitative data analysis (CAQDAS) to simplify the in-depth data analysis process of the textual data collected. Likewise, Sotiriadou, Brouwers, and Le (2014) concluded that using an appropriate CAQDAS to support the data analysis increases the rigor and insights into the data. While Zamawe (2015) recognized the positive influence of a CAQDAS during the analysis process for the researcher, he argued that the researcher must be in control of the analysis. For my study, I identified themes that were relevant to answer the research question from the qualitative data that I collected. As I anticipated a substantial volume of textual data files in the form of transcripts and documents, I used NVivo to support the process of organizing the data files. Additionally, I used NVivo functionalities to aid the coding and themes development process for the data loaded.

The coding approach enabled me to identity codes and categories in the data for the development of themes of my research. Braun and Clarke (2016) depicted coding as a method to search for evidence in the data for the themes. Blair (2015) emphasized coding as a process of repeated coding moving over three stages, starting with open coding, moving to axial coding, leading to selective coding and themes development. Also, Ganapathy (2016) suggested to code in steps, starting with open coding to label the raw data, followed by categorization of the codes to establish a focus of the data, moving to axial coding by establishing relations out of the classifications, leading to the identification of themes. Likewise, Deterding and Waters (2018) presented a three-stage approach for analyzing qualitative data: stage 1 as an organization and indexing the data; stage 2 application of coding and categorization; stage 3 to target the identification and validation of the theme. The use of a process for the analysis in my study aided the analysis phase and enhanced the validity and creditability. Therefore, I used an iterative coding process that started with open coding of the transcripts, followed by axial coding and categorization, and the final phase of themes development and validation of findings. I aided the coding process with the functionalities of NVivo.

The validity and creditability of the findings in my study were positively influenced by achieving data saturation. For the assessment of the level of data saturation, Hennink, Kaiser, and Marconi (2017) argued that code saturation might not be sufficient to assess that the researcher can fully understand the phenomenon. Similarly, Chowdhury (2015) recognized the uncertainty in qualitative research to define the point of sufficient data is collected even when no new data emerges for the identification of themes. On the contrary, Constantinou et al. (2017) observed that themes saturation could be identified by validating that no new shared themes emerge compared to the previous interviews. While member-checking validated the data saturation with each participant, the validation of the saturation of my study required a different approach. Therefore, I used themes saturation for assessing that data saturation in my study was reached.

The process for the analysis started with the organization of the collected data. I loaded the audio-recordings, electronic data, field notes, and reflective journal into NVivo and tagged them accordingly with participant and organization codes. For the transcription of the audio recordings from the interviews and member-checking sessions, I used the third-party transcription service Rev.com. Each transcription, either from an interview or member checking, I validated using the transcription function of NVivo word for word, including immerging into the data. Also, I included any organizational documents that were relevant to my research topic.

Furthermore, I used my field notes and reflective journals that might have guided the analysis process. During the process, I replaced the names of individuals and organizations in the transcripts, including sensitive contextual information with appropriate codes, and loaded the final transcripts into NVivo. I developed the first codes using open coding for the transcribed data in NVivo. I used appropriate functionalities of querying the data, word clouds, word trees, project maps, diagrams, and charts in NVivo to enhance my understanding of the data. Once I reached an understanding of the information patterns and their categorization, I used axial coding to identify relations between the categories.

Additionally, I used the data-code relationship and matrices functionality in NVivo to visually enhance the data and validate the patterns. I derived themes by clustering of categories and their relationships that map to my research question. The themes that I was interested were related to strategies for domain-driven design, DevOps practices in MSA adoption, themes from the literature review, and conceptual framework. For each participant, I reviewed the synopsis of the interview during the member checking session. I used the iterative member checking sessions to validate, correct my interpretations of the transcribed data until no new information was added by the participant. I revised and corrected the codes identified and developed categorizations until no new relevant themes emerged from the data. I used NVivo matrices to validate the thematic saturation of my study. To include any relevant new studies to my research questions, conceptual framework, or literature review, I monitored the literature during the development of my study. The final report contained the themes that were aligned with the research question, literature, and conceptual framework.

Reliability and Validity

Researchers developed studies in such a way to demonstrate trustworthiness, a consistent approach, and a repeatable outcome. Grossoehme (2014) referred to reliability as the degree of how the results of the study can be replicated. Leung (2015) explained the reliability in quantitative research as reproducibility of the outcome and for qualitative research to ensure a consistent outcome. Likewise, Dikko (2016) described reliability when the measurement of the outcome is aligned with the concepts, without bias, less and consistently reproducible. I designed my study to the most possible extent in order to enable increased reliability.

Constructing the right measure of how the answers relate to the research questions was essential for addressing validity in the study. Tuck and McKenzie (2015) argued that validity is an unprecise term in establishing methods for measurement. Consequently, Lub (2015) presented a framework to address validity by linking the verification procedures to the research purpose, paradigm, and perspectives. Similarly, FitzPatrick (2019) noted the importance of using appropriate methods in order to establish trust for the conclusions reached from the inquiry. Dennis (2018) recognized that the researcher is part of the approach in establishing the validity of the findings. Therefore, as the researcher, I included appropriate verification practices in my study to ensure the linkage of my findings to the data collected.

Dependability

The demonstration of the dependability of a study was influenced by the applied rigor, the shown transparency, and the level of documentation. Nowell et al. (2017) reflected for achieving dependability in ensuring logical steps in the research, the transparency of the methods for others, and generating an auditable trail. Similarly, Maher et al. (2018) depicted dependability as the level of detail and audit traces to enable a different researcher to replay the study's outcome. While Morse (2015) argues that the replication of a qualitative study is questionable, he defended the importance of ensuring rigor in the research approach for a high-quality outcome. Therefore, I provided a detailed description of the research protocol, the research design, and the selection of the participants. I established a chain of evidence for the data collection process, the data analysis, the identification of the codes, the development of the themes and findings in the report for a stepwise replication of the results.

Credibility

Establishing confidence in the findings by others is an essential objective of the researcher. Connelly (2016) emphasized that the researcher should employ several techniques such as member checking, extended engagement with participants, peer-debriefing, and reflective journaling to establish credibility in qualitative research. Also, Forero et al. (2018) maintained to include peer debriefing, prolonged engagement techniques for data collection, additional sources of data, field notes, interview protocol, and investigator skills to demonstrate credibility. Similarly, Korstjens and Moser (2018) suggested that strategies for credibility include prolonged engagement, persistent observations, triangulation, and member checking. I got familiar with the case organizations and studied the culture by engaging with the gatekeepers to understand the context and environment of possible participants. While the duration of the interviews was time-boxed, I established a good rapport with the participant and followed up with the member checking. I used member checking, as discussed in the data collection section, to validate the accuracy of my interpretations of the data collected. Also, I employed data triangulation using multiple data sources from interviews and organizational documents. Furthermore, I applied methodological triangulation, as described in the data analysis technique, in using multiple methods of data collection to establish the credibility of my research.

Reaching data saturation is a crucial concern to establish the credibility of the research. FitzPatrick (2019) recommended using appropriate sampling approaches to ensure the researcher reaches data saturation during the data collection. Constantinou et al. (2017) reported that themes saturation could measure data saturation effectively to demonstrate the data is robust for the interview data collected. Fusch and Ness (2015) recognized that the sampling approach influences the required number of interviews and may not be identifiable in advance. Therefore, I used a set of criteria for purposeful sampling of the participants. Additionally, I assessed the data saturation of my study by the evaluation of themes saturation.

Transferability

Detailed and rich documentation of the study allows others to understand the aspects to transfer the findings into a different context. Nowell et al. (2017) depicted transferability as enabling the judgment of others by providing a thick description of the case. Connelly (2016) indicated to provide a detailed description of the environment and subjects investigated to inform the reader of possible transferability of the research findings. Also, Korstjens and Moser (2018) recognized the responsibility of the researcher to enable the reader with a detailed account of the research process and participants to evaluate the applicability of the findings into the reader's context. While the articulation of transferability is not appropriate for this qualitative study, I provided a detailed and thick description for the readers of this research. I provided a detailed account of the contextual setting of the research, locations of the organizations participating in the study, the sampling strategy, the participants, the interview protocol, the data collection process, and scheduling approach, and the data analysis process. The level details may allow another researcher to evaluate the transferability of this research into a different context.

Confirmability

The reduction of the researcher’s bias is an important aspect of qualitative research. Connelly (2016) presented confirmability as the degree of neutrality of the research outcome using methods such as member checking, peer debriefing, notes of research decisions, and audit trails of the research process and data analysis. Forero et al. (2018) depicted confirmability as the confidence in findings by others employing strategies for qualitative research such as reflexive journals and triangulation techniques. Also, Abdalla et al. (2018) concluded that multiple measures must be included in the procedures of a study to reduce the researcher's bias by including triangulation, documenting decisions, reflexivity of the researcher, and evaluation of alternative theories for the findings.

Similarly, Hadi and Closs (2016) illustrated several strategies for supporting confirmability for qualitative studies such as maintaining a reflexive journal and field notes; the use of triangulation to reduce the bias of a single source, single method, or single researcher; the member is checking with the aim of removing bias from the interpretations. I achieved confirmability by using member checking to remove my bias from interpretations of the participants’ interview data. Additionally, I used my reflexive journal to review decisions and identify and remove preconceived notions that might impact data collection, data analysis, development of codes, themes, and report of findings. Furthermore, I used data triangulation and methodological triangulation to mitigate any influence of single data source analysis or single methodological approaches to demonstrate confirmability.

Transition and Summary

In section two, I presented the details of the approach and setting for my study to explore domain-driven-design strategies and DevOps practices that IT product managers and related teams use to reduce the inefficiencies in the MSA adoption in organizations. I approached the organizations and participants in line with the ethical standards of Walden University. To answer the research question, I adopted multiple case studies to identify the best practices employed by the participants in their organizations. The primary source of data was enabled through interviews of participants selected in a purposeful sampling approach. I used a semistructured interview protocol to collect the primary data, including member checking for ensuring the accuracy of my interpretations and supporting data saturation.

Additionally, I took field notes, kept a reflective journal, and collected organizational documents relevant to answer the research question. The organization of the transcribed interviews, notes, and documents was stored in a research database that is supported by NVivo. For the protection of privacy, I replaced the names of individuals and organizations with codes and anonymized any privacy context information. For the data analysis, I employed an iterative coding process that started with open coding of the transcripts, followed by axial coding and categorization, and the final phase of themes development and validation of findings. I used themes saturation for assessing that data saturation in my study is reached. To strengthen the reliability and validity of my research, I used data triangulation and methodological triangulation, reflexive journaling.

I provided a detailed and rich description of the research protocol, data collection, and research report.

Section 3 presents my study's findings, the application to the professional practice, the implications for social change, the recommendation for actions, and future research. Furthermore, I provide my reflections on this research project and the conclusions of the study.

Section 3: Application to Professional Practice and Implications for Change

In Section 3, I present the findings of the strategies and practices for domain-driven design and DevOps used by practitioners to adopt MSA systems. Also, I include the application to the professional practice, the implication for social change, recommendations for action, the recommendation for further research, my reflections, and the conclusion of this study.

Overview of Study

The purpose of this qualitative multiple case study was to explore strategies for the domain-driven design and DevOps practices used by IT product managers to reduce the inefficiencies in the MSA system adoption. The population for this research consisted of IT product managers with roles related to MSA adoption (architects, developers, product owners, software engineers, and platform engineers) of two large organizations (with IT departments higher than 1700 employees) operating globally with one in the customer goods manufacturing industry and another in the banking industry. Both organizations had headquarters in Europe. I collected and analyzed data from interviews and documents following the research protocol. Four major themes emerged: (a) organizational alignment in adopting MSA, (b) ways of working, (c) an experienced-based approach to design MSA systems, and (d) MSA environment landscape. The alignment of the collected evidence to the factors of the conceptual framework contributed to the support of these themes (see Table 1 for the frequency of the conceptual framework factors). The findings of this study showed that strategies and practices used for domain-driven design and DevOps for the adoption of MSA included addressing the organizational complexity with knowledge-driven decisions, flexible ways of working, and evolving support environment.

Table 1

Frequency of Conceptual Framework TOE

	Participants		Documents	
Context	Count	References	Count	References
Technology context	18	391	6	49
Available technologies	18	266	4	19
Current equipment and methods	18	124	6	30
Organizational context	18	316	5	68
Formal and informal linkage structure	18	156	3	20
Communication and top management leadership behaviors	18	140	5	48
Size of organization	7	12		
Slack	6	7		
External task environment	8	16	2	3
Industry characteristics and market structure	5	5		
Technology support infrastructure	4	4		
Government regulations	3	7	2	3

Note. Conceptual Framework based on Technology Organization Environment (TOE); n = frequency.

Presentation of the Findings

The research question of this study was as follows: What strategies do IT product managers use for the domain-driven design and DevOps practices to reduce the inefficiencies in the MSA system adoption? I collected the data through semistructured interviews (*n*=18), technical documentation of the organizations (*n*=2), and publicly available technical reference documents (*n*=5). Participants from both case organizations did not have many documents to share due to the confidential nature of the software systems and proprietary information in internal documents. The analysis of the data was guided by the TOE framework to structure the information provided for the complex and multidimensional adoption approach of MSA systems. I performed member checking (*n*=12) and data triangulation to improve the validity of the findings. The themes were identified by open coding of the transcripts, followed by axial coding and categorization. I used NVivo in version 12 to support the organization of the research database and aid the coding and topic identification. The terminology used for the discussion of the findings is as follows: AP01 refers to Participant #1 of Case Organization A; BP09 refers to Participant #9 of Case Organization B. As topics emerged that were not covered by the discussion in Section 1, I conducted a literature review to identify how existing literature relates to the themes. The next sections detail the four identified significant themes, including minor themes. Each topic presented the strategies used by the practitioners to reduce the inefficiencies in the MSA adoption and related factors of the conceptual framework.

Theme 1: Organizational Alignment in Adopting MSA

The theme of organizational alignment is a concern across the domain-driven design and DevOps areas in adopting MSA systems indicated by participants. The adoption of MSA systems requires an organizational structure that supports the design, development, and operation of such systems within the existing context of the organization. The use of agile software development methods is a typical method used for MSA systems which requires an organization to adopt adjusted development structures. The organizational alignment in adopting MSA also included the alignment of the team and MSA structure and alignment of support for the team as subthemes. The alignment of the organization covered the lifecycle for the MSA systems in terms of design, development, and operations that are necessary to deliver applications to the business.

In this study, 17 practitioners articulated the influence of the organizational structure on the team's ability to design, develop, and operate MSA systems (see Table 2). The importance of the organizational alignment to the teams responsible for the lifecycle of MSA systems was not expected, as the focus in existing literature in the MSA context was primary on technological aspects. Participant AP01 explained, "you restructure your organization around teams that are vertical teams that are driven around the domain." Also, Participant BP05 noted a setup of "development teams that are working in each one of those independent contexts." The concern of the participants for the alignment of the team to the scope of the MSA systems was seen as a crucial aspect of working on the lifecycle activities. Participant AP02 explained a specific organization structure as "we adopted the Spotify model." However, BP03 cautioned that "whole organization needs to be aware of the change of the model." Practitioners must consider the organizational structures in their approaches to ensure the team can efficiently deliver MSA systems with the requested functionalities to the business.

Table 2

Theme 1: Frequency of Theme Organizational Alignment in Adopting MSA

	Participants		Documents	
Major Theme/Subtheme	Count	References	Count	References
Organizational alignment in adopting MSA	17	77	2	8
Alignment of team and MSA structure	15	58	2	7
Alignment of support for the team	8	17	1	1

Note. Theme 1, organizational alignment in adopting MSA; n = frequency.

The theme of organizational alignment in adopting MSA aligns well to the factors of the organizational context (see Table 1) of the TOE framework, namely formal and informal linkage structure, communication, and top management leadership behaviors, size of the organization, and slack because Tornatzky and Fleischer (1990) argued that the formal and informal linking structures of an organization impact the adoption of innovation through the setup of the organization structure and internal behavior. The alignment of the organizational structure is crucial in the adoption of innovation as Participant AP01 emphasized that the structure of vertical teams around business domains is working only when they work on extended scope beyond their expertise and collaborate with other teams to bring software "from the idea creation also to production deployment." Also, Baškarada et al. (2018) argued that significant trust in capabilities must be established within the structure of the organization to allow practical MSA lifecycle support. Therefore, the factor of the formal and informational linkage structure of the framework aligns well with this theme because the setup of the business organization and structure of the IT organization influenced the adoption of MSA systems.

The factor communication and top management leadership behavior of the organizational context of the conceptual framework align with this theme because the setting of the tone and style of communication is essential for the efficient management of the lifecycle of the MSA systems. While not much was mentioned by the participants for top management leadership behavior, the indirect articulation of the setting of the organization structure and challenges provides an indirect insight into this topic. Top-down communication through top management are, for example, security policies related to IT development and policies that require approvals from the top management. Participant AP02 stated that the implementation of security policies requires significant collaboration with the security department to adopt for MSA systems by "sat together with our security department, which again is not directly connected to the development department." The decisions and actions of the top management tend to shape the communication and behavior of the individuals in the organization. The steering of the teams is carried out by the management in a light touch way as Participant AP07 noted: "some organizational problems we try to solve them on a team level or organizational level or technical level to make it easy for the teams." While the freedom supports a mostly autonomous operating team for an MSA system, the delegated responsibility for delivering the MSA systems to agreed qualities is completed with this team as Participant BP02 explained: "our team is fully responsible for the qualification lifecycle surrounding it in production." Thus, this theme aligns well with the factor communication and top management leadership behavior of TOE.

The influence of the size of the organization was minimal within the teams concerned with the MSA lifecycle, due to the small sizing of the MSA teams and independent set up within the organization. A concern by three participants (AP06, AP07, BP01) was the separation of departments in a larger organization that required more effort to align with an MSA suitable team structure. For example, AP06 raised the concern "within this larger organization, everything is new to everyone and especially since there is a separate infrastructure department, operations department, development department, operations is way too far away from development." The low indication by the participants related to the size of the organization suggested that this factor is not a core influencer regarding the practices for the MSA adoption.

The availability of resources, slack in the organization, to support the MSA lifecycle, is another minor concern for the adoption of MSA because only six participants (see Table 1) indicated an influence caused by the slack available in the organization (AP04, AP07, AP09, BP01, BP06, B08). The identification of slack is difficult as the participants did not indicate slack as a topic. However, I considered the nine practitioner questions postulated by Zimmermann (2016) for MSA design concerns as a viewpoint to identify slack. The topics discussed as slack were architectural support, guidance, practices, and community support that is available to the teams to address the challenges in the MSA lifecycle. For example, Participant AP07 stated slack in form of a practice "that the organization has created a specific competence center around microservices to help their teams or you call it squads to get on the journey." Also, Participant BP08 indicated available slack in the organization "if there is any problem, also, there's a community for architects that can help to try to fill the problem." The low frequency of slack indications might be a result of the resource optimized organizational structure. Erich et al. (2017), and O'Connor et al. (2017a) referred to agile methods as it enabled to create small and nimble organizational structures to handle the software lifecycle. Thus, as the MSA teams operated as independent as possible, they did not rely on other available resources for their activities, which supports the weak alignment with TOE's organizational context factor slack.

The topic of MSA and organizational impact seems to be of limited interest in the research community so far. Leite, Rocha, Kon, Milojicic, and Meirelles (2020) argued that the literature is incomplete for developing decisions to restructure an organization towards a suitable structure to adopt DevOps. However, Baškarada et al. (2018) reported the concerns of practitioners that the typical organization structure as of plan, build and run is not suitable for MSA teams that operate across these organizational silos, which tightly aligns with the findings of this theme as Participant BP03 explained "from the definition phase to the execution. Nobody else but the team is involved in the lifecycle of the software." Case Organization A employed a specific organization structure as Participant AP08 reported to be organized "in the Spotify model and there is still functional groups of interests" which aligns to the research from Smite, Moe, Levinta, and Floryan (2019) which presented how the organization model of Spotify fosters alignment, knowledge sharing and decision making in an agile software development process. BP05 presented the aligned development team set up for the independent context as an advantage in the management of the MSA lifecycle as "that those teams can then concentrate on that specific domain." Contrarily, BP03 described the difficulties of people in their mindset as most challenging part to transform into the new MSA aligned team structure. However, the statement by Participant BP03 aligns well with the research by Zhu et al. (2016) arguing that the adoption of DevOps and MSA is not a simple and straightforward process which places significant stress on the organization.

Practitioners tend to consider organizational structures in their approaches to be able to successfully adopt MSA systems. The variations of organizational structure impact the practices and extend for domain-driven design and DevOps. In both case organizations, participants worked in different teams concerned with the MSA adoption and therefore implemented variations of practices. While participants referred in the Case

Organization A to the Spotify model as implemented organization structure, Case Organization B seemed to naturally be evolved into an MSA aligned structure using the DevOps construct. However, participants of both case organizations experienced in different flavors the challenges to fully align with an efficient organization structure for the MSA construct.

Subtheme: Alignment of team and MSA structure. Fifteen participants indicated that the alignment of the team to the organizational structure for the adoption of the MSA construct is a significant concern for the complete lifecycle (see Table 2). The alignment refers to the organization that designs, develops, and operates the MSA systems to the business organizational structure. The participants highlighted the complexity of decision making and impacted for design, deployment, and operations when the fabric does not support the ways of working. For example, AP06 noted the challenges of a siloed structure for the implementation of a feature as “the back end needs to do this first and that they can fit something and then for two years, nothing happened, and then the front-end team should finish the feature.” Also, Participant BP02 expressed the requirement of an aligned organizational setup as “you need to have a proper organizational setup for this. It's not going to work if some guy builds some library and forget about it, and there is no one maintaining this library. It's going to be dead quite soon.” Thus, the complexity of an organization can’t be ignored by a practitioner during the development and operation of software systems. The complexity and activities vary over the software lifecycle processes such as requirements analysis, planning, design, development, testing, deployment, and operations of MSA systems.

In this study, 15 participants indicated that the business structure informs the team on the boundaries of the business domain and dependencies on other units of the organization. The structure and alignment of the organization supporting the MSA design, development, and operation were articulated as a concern to manage the scope of MSA. Participant BP09 noted how to establish an aligned organization structure by “split logics so they would have their service for this team, another service for this other team and one for service for this team,” which aligns with the finding by Pautasso et al. (2017) that the domain-driven design allows to split the business area into a bounded context and to establish a team around this bounded context. However, the organizational structure might not be the same as for the bounded context of the MSA system as Participant BP05 mentioned the alignment could influence the MSA team setting in “how we approach domain-driven design is the fact that in some organizations which are multinational organizations that have development teams distributed geographically.” Thus, increased the challenges for the team as going across organizational boundaries of the structure. Creating an independent team seems to be a concern of Participant AP09 as “we are starting with just teams that try and build new services that do not exist yet” to mitigate the implication caused by the organizational structure and aligns with the findings of the research by Kuusinen et al. (2018) that the organization structure can become a major blocker in adopting DevOps. Also, Hasselbring and Steinacker (2017) recommended that the team responsibilities should be aligned to the business capabilities of MSA systems for efficient lifecycle management. The arrangement of the team to the MSA construct was an essential consideration to enable a successful MSA adoption. Establishing a team outside of the bounded context will increase the inefficiencies of the team designing, developing, and operating the MSA systems. While participants are concerned with the structural alignment, the team's internal structure has to fit as well to the MSA context.

The boundaries confine the MSA construct but do simplify a complex system for a team. Participant BP05 described the approach to define the team based on the bounded context of the business for MSA systems as follows:

> Applying Domain-driven design is that you can slice the development of let's call it a complex system into modules that are attached to a concrete business context. And so, you can set up teams of development teams that are working in each one of those independent context.

Soldani et al. (2018) argued that the challenge of MSA is the intrinsic complexity of such a software system. Also, Taibi and Lenarduzzi (2018) referred to MSA patterns that present complexity at various levels in general for the following areas: (a) API gateway, (b) service registry, and (c) data management and storage. The complexity of the MSA must be handled by the team in charge of the lifecycle with the right level of structure and skills. Enabling an appropriate inner structure to address the scope was crucial for the practitioners. AP02 presented the use of the Spotify model as a suitable team structure for enabling MSA work by "sitting in the same area with my product owner, and my business analyst, and my fellow developers working on the same service where we need to improve." Smite et al. (2019) described the Spotify model to share knowledge and cultivate a strong passion for innovation. Knowledge sharing and helping to develop the individuals were part of the practice effort. BP03 strongly supports that the ownership of the complete lifecycle relies upon the team in charge of the MSA system as expressing "from the definition phase to the execution. Nobody else but the team is involved in the lifecycle of the software." Similarly, Hasselbring and Steinacker (2017) recommend enabling the responsibilities and activities can be covered within the team for the lifecycle activities. Participant AP02 stated that it is "important to be able to let the people independently work," which should be supported by the organization structure. Also, Cuesta, Navarro, and Zdun (2016) considered MSA as a construct with key aspects of the independence of the team to carry out work. Even Suram, MacCarty, and Bryden (2018) presented MSA as a way to enable the independent design, development, and consumption of the software. Keeping the responsibilities within the team and allow self-organization is part of the principles of agile methods. Practitioners were concerned with efficiency, so setting up environments in the most suitable way for the team members to avoid them getting blocked in their work and also limiting the complexity of work to address by the team. While the alignment of the team structure is one concern, the composition of the team adds another dimension to the organizational structure.

Practitioners consider the team composition part of the success. According to six participants, the team structure should allow the team to work independently on the MSA system in delivering new functionalities to the business organization. Five participants recommend including particular skills into the team to work on the business and technological aspects independently. Participant AP07 suggested creating "kind of full-stack teams, where you have a team which could, in the end, deliver a feature on their own without having to interact with other teams a lot." More specific was Participant

BP05: “we have deployed SRE teams that are following that operations framework” to enable “the understanding of availability, the understanding of performance for those microservices.” Similarly, AP02 stated, “business analysts have, actually, a very technical background, which is the most beneficial part because those business analysts are requirements engineers” in our teams. Establishing a skill within the organization to foster the distribution was indicated by Participant AP04 to have an “initiative to also enable there a DevOps that will be there as a person as well, and also as a role. And that this role will also tag then topics like the implementation of pipelines.” Also, Baškarada et al. (2018) identified the need for relevant skills at an advanced level of distributed systems and DevOps to master the complexity of MSA systems. Participant BP02 explained the scope of architecture responsibilities as “for deciding things like that. So basically, when I'm considering whether they should be extracted into the library, a certain functionality, where it rather has to be duplicated in the code.” The need for experienced individuals with a broad range of technology and system understanding was not easy to fulfill. Finding such qualified personnel posed another challenge for management. Therefore, creating a suitable inner structure with clear responsibilities is easing the pain towards improving the efficiency of the team.

The focus of the participants on the alignment of the team to the MSA systems including the identification of boundaries and needed skills aligns well with the organizational context of TOE (see Table 1) because without the alignment between team and MSA system the lifecycle activities would result in inefficiencies during the MSA adoption. Participant BP05 noted the aim for arrangement within the team structure as “the balance between accountability and our autonomy,” which aligns well with the formal and informal linkage structure of the TOE framework because accountability and autonomy must be agreed within the organization and shared by the team.

The finding of this subtheme alignment of team and MSA structure aligns with the literature regarding the research on large scale agile software development, because MSA system development is influenced by agile development approaches (Leite et al., 2020). Alignment between the organization and the software development organization is crucial to deliver the right software product. Fuller (2019) argued that a functional organizational model can be harmful in a software development environment as it is impacting the behavior and effectiveness during the development, which was indicated by the explanation of the partition as creating boundaries between the team members. Similarly, Kuusinen et al. (2018) argued that organizations need to remove organizational silos to adopt DevOps, which aligned to the effort of the participants to create a group that can work independently and autonomously on the lifecycle of the MSA system. Also, the research of Dekoulou and Trivellas (2017) asserted that creating an organization structure with a high degree of self-control and limited direct guidance fosters the adoption of innovation, which was consistent with the concerns of the participants. Considering that agile methods are working best when autonomous teams design, develop and operate the MSA systems over the software lifecycle. While the focus of the team is based on addressing the business needs by adopting MSA-based systems, the supporting components such as DevOps toolchain, proven organizational solution, knowledge, and underlying infrastructure components of the MSA platform are outside of the MSA teams' responsibilities.

Subtheme: Alignment of support for the team. Eight participants indicated that the alignment of support for the team fosters the adoption of MSA systems (see Table 2). Supporting the team in the activities required to address the lifecycle of the MSA with knowledge, decision guidance, re-usable enterprise assets, and platforms are part of the improvement culture in an organization. Participant BP05 recommended to split the responsibilities for the MSA support environment needed for MSA systems from the MSA as follows:

> So the skills of DevOps toolchain and platform for deployment and the skills of the software developers that are actually implementing the microservices and that have to invest more time in understanding the business domain and the vocabulary and the processes behind that. So, I think it's a good practice to split those responsibilities and having a platform team that is taking care of the, let's say, the platform related topics.

Also, Participant AP08 stated that "the platform team is actually responsible for making the process of deploying to the cloud as easy as possible for developers." The scope included establishing a common approach for the platform services as indicated by Participant BP08, for example, "the names of the PODs in the Kubernetes clusters or things like that. But this evolves to provide support to the teams this a further team to do that." The focus of the MSA team was on the business value, not on the underlying infrastructure components. While the choice and decision of tools and platforms were with the MSA team, the organization made appropriate support available for these technologies. Besides the technology that was used for the MSA systems, architectural knowledge is required.

The design of MSA systems requires experienced architects and engineers. Participant AP09 explained the architecture support for the MSA team “basically those decisions are done by the so-called architecture team, there is a dedicated team in our organization that deals with those architecture topics.” Also, Taibi and Lenarduzzi (2018) recommended using software architects for system-level decisions to mitigate the challenges of MSA systems. These architectural decisions, as mentioned by Participant AP07, for example, “we give recommendations for synchronous versus asynchronous communications” between MSA systems. Architecture support was required in the process of architectural design decisions. The architectural support was called in on-demand and was not mandated by the organization to support the MSA team in their effort to create value for the business, as explained by Participant AP07. The decision to engaging these support teams was with the MSA team itself.

This subtheme aligns with the TOE framework in the organizational context because of the importance of having an appropriate organizational structure in place to adopt MSA systems efficiently. Eight participants indicated that splitting the responsibilities between the teams with the MSA business focus and architecture, platform, and DevOps tools was a good practice, which was consistent with the factor of the formal and informal linage structure of the TOE framework. Participant BP05 recommended this as follows:

> It's a good practice to have a specific team that is in charge of maintaining the toolchain, and the platform with services are then developed or deployed with Amazon web services or on-premise solutions or whatever that is. So, not their teams that are implementing the microservices, but to have a separate team that is in charge of maintaining the CI/CD pipeline, the monitoring tools, the testing tools that all teams are, that the different teams are using, that are maintaining the let's say the cloud environments that are being used for the deployment.

Providing an organization structure with split responsibilities that allowed MSA teams to focus on the business value creations and adoption of innovations aligns well with the focus of the organizational context of the TOE framework.

The findings of this subtheme were partially covered in the existing literature only. While DevOps tooling and MSA technology topics were identified, no discussion was identified if a support structure for MSA concerns such as container platforms, DevOps tooling, architecture expertise is required or beneficial. Leite et al. (2020) argued for DevOps adoption that creating a DevOps team is a silo and considered not best practice, without making a recommendation on how to develop and maintain the DevOps tools in the best manner. Also, Zimmermann (2016) listed lightweight container usage, continuous delivery, and DevOps as a concern for the MSA that require support, which aligns with participants’ answers.

The supporting environment for MSA consists of many tools, platforms, and knowledge pools. Support was provided for these tools from the organization to ensure usability and availability for their environment. Knowledge pools in the form of architecture and engineering support were part of the environment of fostering MSA adoption. The case organizations created an environment that allowed the MSA teams to focus on the business demand by providing a supporting structure for the guidance of problems, helping with proven enterprise solutions, and platforms that are improving the efficiency of the MSA systems.

Theme 2: Ways of Working

The theme ways of working represent the importance of how the team was engaged in collaboration, knowledge acquisition, and knowledge application. The focus on the ways of working related to the activities that were required to design, develop, and operate MSA systems with the requested functionalities and availability for the organization. There were three sub-themes in the ways of working that influenced how teams work with MSA systems. The first topic was around the guidance that is available for the team working on MSA systems, the second is related to the concern of how to foster the collaboration, and the third topic is how to encourage the acquisition of knowledge. The ways of working included practices for domain-driven design, DevOps, and unique approaches of the participants to address the challenges of the MSA lifecycle activities.

All 18 participants discussed practices of how individuals interact during the process of the design, development, and operation of MSA systems in the organization (see Table 3). The focus of the participants on the ways of working was expected as practitioners required to address the challenges that emerge during the stages of the lifecycle of the MSA-based systems (Soldani et al., 2018). Participants indicated that the setting of the team plays a vital role and impacts the knowledge dispersion required for MSA systems within the members of the team. Participant AP02 explained that it is good “to have self-organized teams” that allowed to “keep all the knowledge needed within one team.” Participant BP06 noted to benefit by “actually standardize how do we want to do operations across those teams.” While AP03 stated that “we provide a standard, which is easy to be used, but we don't force them to use it.” However, BP03 cautioned that if no standardization was embraced, “teams can do whatever they want in their microservices.” Zimmermann (2016) argued that for a successful MSA implementation, a combination of pattern and modern software engineering practices is required.

Table 3

Frequency of Theme Ways of Working

	Participants		Document	
Major Theme/Subthemes	Count	References	Count	References
Ways of Working	18	188	6	65
Guidance for teams working on MSA	18	104	5	38
Foster collaboration as community	15	49	4	15
Encourage to acquire knowledge	11	35	4	12

Note. Theme 2, ways of working; n = frequency.

The theme ways of working align to the TOE framework in the area of technology context because of the technology choices made, the reuse of existing deployed technologies, and proven approaches in the organization used by the teams to adopt MSA systems. Chandra and Kumar (2018) recognized the choice between internally available and external technologies as influential for the adoption of innovations, which aligns with the focus of the participants on ways of working and their decisions in the areas of domain-driven design and DevOps during the activities of the handling the MSA systems. The influence of available technologies on the ways of working that participants of this study discussed supported the factor of available technologies of the TOE framework. The technology choices made by the participants defined the ways of working for the team and have subsequent implications on existing technologies in the organization. Similarly, Oliveira and Martins (2011) described the interlink between technology that is available and the existing internal technologies, including processes for the adoption of innovations for the technological context. The participants if this study discussed the considerations of the in the organization available set of technologies and practices for the ways of working which supports the factor of current equipment and methods of TOE. Furthermore, participants indicated the openness in selecting appropriate technology and the flexibility to adapt a domain-driven design approach and DevOps practices as a significant aspect for the lifecycle of MSA systems. The theme ways of working is the focus of practitioners considering the impact of technology choices either new or existing on available internal technologies and processes to efficiently achieve the delivery of new innovative MSA-based functionalities to the organization. Thus, the theme ways of working align well with the technological context of the TOE framework for adopting MSA-based system in an organization.

The discussion on ways of working was present in the existing literature in the form of recommendations for designs and implementations, case studies, and research regarding challenges of MSA systems. Soldani et al. (2018) investigated reported pains that exist over the lifecycle of MSA-based systems by practitioners that inform processes and approaches, which tightly aligns with the finding from this theme of ways of working. During the design phase, domain-driven design decisions are needed which inform the development phase with subsequent decisions. These decisions were necessary to customize the DevOps practices and the selection of suitable platform choices. Practitioners tended to consider the influences of decisions overall stages in their ways of working to support the efficient delivery of MSA systems to their organization. Also, Taibi and Lenarduzzi (2018) reported elven bad smells in MSA-based developments that require teams to establish particular practices ways of working to overcome the challenges, which were discussed by the participants of this study as well. The technical difficulties that MSA systems pose for practitioners resulted in practices that inform the ways of working in organizations leading to a set of “best practices” in their quests of improving the inefficiencies of MSA systems. While the research of Taibi and Lenarduzzi (2018) focused on technical and architectural aspects, the study by Soldani et al. (2018) provided an investigation of the practitioners’ concerns over the lifecycle of MSA systems, which both are aligned with the ways of working of this theme. Practitioners do consider multiple aspects in their ways of working for MSA-based system to provide (a) guidance to the teams to enable that the systems can be managed by the organizational capabilities, (b) sharing of knowledge across the teams in the organization to support the evolvement of the strategies and practices, and the (c) elevation of the individuals knowledge to make better decisions to reduce the inefficiencies of MSA-based systems.

Subtheme: Guidance for teams working on MSA. All participants referred that guidance for the teams is available in the form of flexible best practices, re-usable artifacts, and expert recommendations. Participant BP09 noted for the development activities a predefined set of DevOps practices “we have set up tools that is working across the company” acting as the development framework to “like align all the projects together.” Also, Participant AP06 stated, “to come up with a technical foundation and with guidelines of how to make such a microservice architecture possible” is increasing the efficiency of the team. Contrarily, Taibi and Lenarduzzi (2018) reported that too many standards might cause a shortage of knowledge in the event of too little adoption of the standards in the organization. The effort in developing guidelines and standards is a typical quest of organizations to ensure the availability of sufficient capabilities to handle the set of technologies efficiently. Participant AP04 noted as an example, the platform team “gives out the rules on that, and our DevOps team applies this rule” to ensure consistency for the operational usage of the platform. The finding of rules for the DevOps environment aligns with the outcome of the research from Gill, Loumish, Riyat, and Han (2018) as DevOps teams should focus on delivering software and not dealing with platform issues. Participant BP01 emphasized the focus on particular approaches in the DevOps pipeline as “it's really important how we use the repositories and the runs strategies.” Participant BP02 went even further in demanding “to have the same way of, the same format of logs, same way of doing monitoring and technical alerting, having the same dashboard or an ability to build the same dashboards” to simplify the maintenance and knowledge generation of MSA-based systems. More explicitly, Participant AP01 described the guidelines to “explain step-by-step, how actually, not only to deploy” but also to elaborate “often some processes” as required in the organization. The focus on a defined way of working was as expected, as Ebert et al. (2016) recommended deploying a tailored DevOps strategy for MSA-based systems and evolve over time with the knowledge gained. As well as Participant AP07 suggested, “you need to give them tools in their hands” to streamline the operational expertise for the MSA deployments. Similarly, Balalaie et al. (2016) recommended to limit the choices of technologies and provide templates to ensure the organization can support the MSA-based environment. Practitioner tended to develop guidance for reoccurring issues in the development so that the team and induvial can be relieved of an elaborative decision-making process to evaluate the best options in the context of the organization.

Only a limited set of guidance for the design of MSA systems is discussed and made available. Interestingly, the design area of enabling access to MSA-based applications using an application programming interface (API) is an area that is actively guided as Participant AP02 stated: “we're also building API guidelines so that the microservices are able to communicate between each other on a more or less standardized way.” Participant BP04 focused on the efficiency of the work required “if I expose it to API and it, have events, I am able to share the data without much effort.” Guidelines enable us to increase efficiency in the way of working as Participant BP05 noted that a team “should be able to support more than one API because all of them look the same in terms of the toolchain.” The concern seems to be linked to support the interoperability between the MSA-based systems by creating efficient ways of working for the teams that design the API pattern as Zhao, Jing, and Jiang (2018) argued that an API is an essential component of any MSA-based systems. Therefore, having guidelines available on how to address the technical issues allowed the participants to establish efficient ways of works for the team.

The subtheme aligns with the TOE framework in the technology context because of the importance of guiding the decision-making process of available technologies, existing assets, and existing methods in the organization for the adoption of MSA-based systems. All participants indicated that providing a set of guidance supported the team in the decision-making process of designs, development, and operations, optimizing the ways of working. As noted by Participant AP08, “we are creating handbooks” so that the teams “can actually follow these guidelines.” Practitioners tended to create guidelines for the organization when complex situations require proven solution approaches to increase the success rate as Awa, et al. (2017b) identified that the complexity of sophisticated technology could impede the adoption in the organization. Supporting the teams with guidelines that addressed emerging issues and challenges with proven solutions during the lifecycle of MSA systems aligns well with the TOE framework.

The findings to establish guidelines are similar to the recommendations from existing literature such as from Ebert et al. (2016) and Balalaie et al. (2016) to develop guidelines for engineers to create ways of working for their MSA-based construct. While the participants discussed the development of guidelines for particular areas, the extreme case of a complete set of guidelines as a framework for MSA systems as such as the unicorn framework developed by Pallis et al. (2018), was not envisaged. Similarly, the experience report by Bucchiarone et al. (2018) presented several technical aspects as input for a guideline in the migration from a monolith to an MSA-based system that would require the consideration of a particular context in the directive of the organization to be useable. Furthermore, Larrucea, Santamaria, Colomo-Palacios, and Ebert (2018) emphasized as well to create tailored guidelines as the technical choices, framework selections, design patterns, software languages, and tooling depend on the organizational setting. Considering that organizations typically have multiple MSA systems with different technologies and approaches under development, their leaders may benefit from creating guidelines that simplify the choices to be taken by their MSA teams for proven solutions for particular challenges in the lifecycle of MSA systems.

Subtheme: Foster collaboration as a community. Fifteen participants indicated that collaboration is an essential attitude and setting while working with MSA systems. While participants presented multiple topics and strategies in the lifecycle of MSA systems, there was an underlying agreement of the importance of active collaboration within the team and across the organizational silos to adopt MSA systems successfully and efficiently. Participant AP01 noted that "you have to bring not only but also both management, development, and all others in the company together." More specifically, Participant AP02 referred to intensive collaboration as "the requirements engineer from the business side" engages with "the experienced developers" and "work pretty closely together to design" and "sitting closer to our operations team." Also, Participant BP03 stated, "We collaborate with some other external teams" and "we do a lot of knowledge transfer to those teams." The exchange of knowledge as a critical aspect of collaboration was highlighted by Participant BP08 "because we are a small team and it does not make sense to have that specific expert center" instead establishing "a community with shared information, with all the individuals" as an essential approach for the adoption of MSA-based systems. The use of the community was noted by Participant BP09 as an excellent option to collaborate on solutions for problems as "they can comment," "they have more experience," and "they will know, or maybe they can relate to the issue and recommend different approaches." Six participants (AP03, AP07, AP09, BP06, BP08, BP09) indicated the model of the community is a valuable collaboration approach in their domain-driven design and DevOps work. The importance of collaboration while working on the lifecycle of MSA systems should be considered in the ways of working by the leading manager so that the individuals in the team can get ideas and support to solve the problems during the design, development, and operations.

The subtheme to foster collaboration as a community supporting the team in addressing the challenges in the adoption of new technology and integrating with existing solutions and methods used by the organization as mentioned by participants aligns well with the technological context of the TOE framework. While participants referred to the technical challenges, they experienced in the lifecycle handling of MSA-based systems, they used existing available methods from the community to solve their problems, which aligns with the factor of current equipment and methods of the TOE framework. As Participant BP09 noted, "we have a community" and "we show different ways of doing things, different improvements" which was an indication of using current solutions and methods for adopting MSA systems, which aligns well with the TOE framework. While Erich et al. (2017) identified that DevOps required organizations to foster collaboration, there was no indication if in the form of community spirit. However, Sorgalla, Rademacher, Sachweh, and Zündorf (2018) presented a vision to improve the cooperation of the teams during the MSA development using a conceptual model to consult other MSA models developed. Practitioners needed to solve problems efficiently, therefore, the emphasis of the participants in fostering a community spirit to ensure collaborations of best practices for domain-driven design and DevOps in the adoption of MSA systems aligns to TOE.

The findings to foster collaboration as a community aligned to the existing literature for agile software developments as participants AP07 and BP06 referred to agile software methods in this context. In this study, participants indicated to use a new form of structures such as guilds and community of practice as means to improve collaboration for teams working on MSA-based systems, which aligns to the research by Smite et al. 2019 that new generation software require new structure for collaboration such as developed by the company Spotify. Hekkala, Stein, Rossi, and Smolander (2017) identified that open communication and meetings are important for increasing knowledge, which aligns with the theme of foster collaboration communities. Uludag, Hauder, Kleehaus, Schimpfle, and Matthes (2018) suggested an overall framework and setup for supporting a large scale agile development with domain-driven design at the core that addressed the collaboration at the domain-driven design level without addressing the overarching collaboration need of the development teams in finding community-driven solutions. However, the findings of Baškarada et al. (2018) included several challenges in the development, organizational culture, and organizational structure for the adoption of MSA-based systems but did not make any specific recommendations for collaboration to overcome the organizational issues. Contrarily, the recommendations by Ebert et al. (2016) that DevOps teams closely collaborate with other teams to continuously improve the ways of working, which aligns with the findings of this theme. While different organization structures exist, IT managers should consider establishing a community-based collaboration so that MSA teams can improve their MSA systems and contribute to the practices for domain-driven design and DevOps.

Subtheme: Encourage them to acquire knowledge. In this study, 11 participants emphasized that creating, gaining, and improving the knowledge acquisition was part of the MSA lifecycle activities (see Table 3). Nine participants indicated knowledge acquisition on the levels of individual knowledge, MSA team knowledge, and outside team-based knowledge as part of the ways of the working theme. Regarding the induvial knowledge acquisition, AP02 mentioned that "we also had a training session, where developers gave business analysts and product owners a bit more insight what they do from a technical point of view." Alternatively, Participant BP01 referred to hire the knowledge into the team by "having this skill set on the candidates." Similarly, Participant BP08 indicated that "we hired specific profiles" to speed up the DevOps adoption. Having an end to end knowledge in the team is crucial for agile software development methods (Waardenburg & Vliet, 2013), which was a concern by the participants. While the induvial knowledge level was important in contributing to the team's performance, the team itself must gain, create, and establish knowledge of the MSA systems.

The increase of teams' knowledge typically happened via internally organized training sessions that cover the particular construct of the MSA systems. Participant BP02 described conducting two hours of bi-weekly chaos engineering sessions being the smartest way "not only to find what is not working or what can be broken in our solutions but also boost the expertise of the team." Also, Participant BP05 noted that to conduct "specific trainings around what does it mean to deep dive into a bounded context, what is the ubiquitous language" to increase the team's knowledge on the domain-driven design and business domain needed for the MSA system. The team's self-regulated learning is affected by the day to day workload and urgency of matters to improve the knowledge of the software systems under their responsibility (Annosi, Magnusson, Martini, & Appio, 2016), which was not indicated as a motivational factor by the participants. While enabling a particular knowledge of the MSA systems under the lifecycle management by the team is one aspect, the other aspect is to increase the knowledge of the team regarding the strategic organizational topics.

The organization is steering the overall knowledge acquisition of the team by creating large training programs for many organizational functions. BP02 noted, "we used to have DevOps tournaments in the organization" for all platform engineering squads to increase the overall knowledge of DevOps using a competitive approach. Also, Participant AP01 stated, "there's a lot of retraining of the squads involved" as part of the team-based knowledge acquisition approach for DevOps oriented infrastructure. Smaller scaled organizational training was mentioned by Participant AP09 "in the case of microservices; this training would be done by the backend chapter leads," and it "happens in a different context in our organization." Supporting the development of the organization in their knowledge acquisition approach using an appropriate organizational structure (Smite et al., 2019) was indicated by the participants as working well. As the structure of the organization influences the knowledge acquisition of MSA teams, IT managers should consider setting up structures and approaches that support the knowledge requirements of MSA-based software systems.

Adopting MSA systems is a complex undertaking for many organizations as it requires infusing a significant amount of knowledge into the organization, teams, and individuals involved in the adoption. The participants' focus on knowledge acquisition is aligned to the TOE framework (see Table 1) as the knowledge management positively influenced the adoption of innovation (Soto-Acosta, Popa, & Martinez-Conesa, 2018). BP02 realized that "to educating people" on the topic of domain-driven design to increase the expertise within the team, which aligns with the technological context of the TOE framework for sophisticated technologies such as MSA systems (Awa et al., 2017b). Furthermore, the TOE framework emphasizes on the technology competence; the increase of the knowledge for domain-driven design strategies and DevOps practices increased the wisdom in adopting of MSA-based systems; this was consistent with the technological context of the TOE framework.

Knowledge of the technical aspects was a crucial component in the lifecycle of MSA-based systems that align with the findings in the existing literature of insufficient skills and deployment complexity for the MSA adoption by Knoche and Hasselbring (2019). Also, Zimmermann (2016) posed questions by practitioners to increase the knowledge of the adoption of MSA systems in organizations. Interestingly, Annosi et al. (2016) stated that agile methods tend to create time pressure for software development teams to frequently deliver software improvements and decreasing the ability of the individual and teams to acquire knowledge, which was not indicated to be an issue by the participants. While the participants referred to domain-driven design, DevOps, and platforms as crucial areas of knowledge, Zimmermann, Miksovic, and Küster (2012) developed a knowledge management system to structure the areas of knowledge concepts in different areas. The investigation by Soldani et al. (2018) focused on understanding the challenges of the MSA-based systems that allowed to identify the knowledge deficits over the design, development, and operations lifecycle, which is consistent with the concerns of the participants. The context of the required knowledge is different for each organization and MSA system that IT managers should consider in developing strategies for customized knowledge acquisition for individuals, teams, and the organization involved in the MSA system adoption.

Theme 3: Experienced-based Approach to Design MSA Systems

The theme experienced-based approach to design MSA systems presented the considerations for the design decisions and choices required for MSA systems. The focus of participants around the practitioner-based domain-driven-design concern was addressing the design phase of the MSA system. There were three sub-themes in the experienced-based approach that linked to the design aspect of the MSA systems in the organization. The first topic was about how to establish the boundaries or the cutting of the MSA system, the second was regarding the use of design patterns that enhanced the domain-driven design, and the third topic was how to enable access to data sources of MSA systems using data management systems. The experienced-based approach included unique approaches of the participants to address the challenges experienced for the design of MSA systems.

All participants discussed practices related to the design of MSA systems that worked for their organizational information system environment (see Table 4). The focus on experienced-based domain-driven-design decisions by the participants is expected as a multitude of various challenges for MSA-based system exists (Jamshidi et al., 2018; Rademacher et al., 2018). All participants mentioned strategies and the associated difficulties in identifying the boundaries of the domain to align with the domain-driven design approach for MSA systems. For example, Participant AP02 referred to “isolate standalone domains in the monolith and create microservices out of them” as a crucial task for transforming existing monolithic software systems into MSA-based systems. The other topic articulated by fifteen participants related to the use of design patterns to ease particular problems of the MSA that are not addressed by the domain-driven design approach or pose other challenges. As Participant BP04 noted, “to circumvent the problems of having 20 developers in one service, I went with the strangler pattern.” Boundaries for MSA systems can exist in the business domain and as well in the data layer that should be considered for the design. Accessing data that is stored in another MSA is a common concern requiring a careful design decision to avoid performance impact as indicated by eight participants. Participant BP05 stated that “each bounded context should come up with their data model” and enable the best-suited data access.

Finding the right split for the domains and data, including allowing access, was crucial for MSA systems (Taibi & Lenarduzzi, 2018) and an area of concern for the participants. Designing MSA-based systems required significant knowledge and experienced team members by all in the lifecycle activities involved to decide the best solutions and approaches for the development.

Table 4

Frequency of Theme Experienced based Approach to Design MSA Systems

	Participants		Document	
Major Theme/Subthemes	Count	References	Count	References
Experienced based approach to design MSA systems	18	186	5	34
Establishing the boundaries of MSA	18	81	2	9
Design patterns enhancing domain-driven design	15	57	4	20
The importance of data management	11	48	4	5

Note. Theme 3, an experienced-based approach to design MSA systems; n = frequency.

The theme experienced-based approach to design MSA systems aligns well with the technological context of the TOE framework (see Table 1) because the organization required in-depth knowledge of the technical design decisions that MSA systems need including proven options to integrate with existing software systems of the organization. MacLennan and Belle (2014) identified for the technology context of TOE the use of standards and platforms including the compatibility need with existing enterprise architecture and infrastructure as relevant for SOA adoption, which aligns with the focus by the participants on design patterns, data access and enabling the right boundaries for the MSA systems. Similarly, Wang et al. (2016) found that the compatibility with existing systems eases the adoption approach of new software systems, which aligns with the attention of the participants regarding the identification of boundaries in their existing monolith and the selection of proven design patterns for the design of MSA systems. The focus of the participants to select and adopt new technologies for their MSA-based applications support the factor of available technologies from the technological context of the TOE framework. Similarly, the emphasis of the participants regarding the integration requirements between existing applications and the MSA system in selecting proven technologies of the organization and appropriate design patterns support the factor of current equipment and methods of the technology context. The participants tended to integrate new technology and existing available technology into their approach naturally, as Participant AP01 explained their decision approach on enabling access to the MSA system to "not communicate to each other directly through API calls but we tried to move to an event-driven architecture." The theme experienced-based approach to design MSA systems was the emphasis of the participants on the available technology assets that are available for use and current technologies deployed, including their organizational methods that need to be considered for the design of MSA systems to deliver the required functionalities to the organization. Therefore, the theme of an experienced-based approach to design MSA systems aligns well with the technological context of the TOE framework for adopting MSA-based systems in an organization.

The theme of an experienced-based approach to design MSA is available in the existing literature through experience-based migration reports from monolithic software systems to MSA case studies, research on particular MSA-based solutions, and design studies for MSA and domain-driven design approaches. The experience report in migrating from a monolithic application to an MSA-based system and the requirement to solve particular architectural monolithic problems in MSA (Bucchiarone et al., 2018) aligns well with the findings from this theme. Similarly, Knoche and Hasselbring (2018) presented a case study of decomposing an existing software into MSA systems, including detailing the modernization process that aligns with the focus of the participants in moving from monolithic software system to an MSA-based system. Organizations want to have systems that are simpler and quicker in adopting to new business requirements (Soldani et al., 2018), which imposes the migration from a monolithic systems to MSA-based systems for the organization. Also, the finding of the industrial inquiry on MSA-based systems from Zhang, Li, Jia, Zhong, and Zhang (2019) identified challenges with boundaries of microservices, data management, and design patterns as topics, which tightly aligns with the findings of this theme. While Fritzsch, Bogner, Wagner, and Zimmermann (2019) reported only one pattern and nothing on data access as the finding of the microservice migration industry study, participants of this study used several other design patterns. They focused on multiple means to provide access to data for their MSA-based systems. Patterns allowed the practitioners to address problems in an efficient way of predefined approaches for the design of MSA-based systems, which was repeatedly mentioned by the participants. The significant variations of existing software applications in an organization will require practitioners to evaluate the best options for the design of the MSA system to defining optimal boundaries for the systems, appropriate select pattern to simplify the MSA, and identify the most effective way to provide access to data for the MSA systems.

Subtheme: Establishing the boundaries of the MSA. All participants indicated the importance of establishing appropriate boundaries of the MSA system to support the optimal size for the environment (see Table 4). As Josélyne, Tuheirwe-Mukasa, Kanagwa, and Balikuddembe (2018) identified, the bounded context captures the business details of a single domain so that the services can be organized in appropriate sized MSA systems. Thus, 12 participants discussed the identification of the boundaries as a critical step to enable the right size of the MSA systems for the team. Participant AP04 explained the approach to establish an appropriate size for an MSA by “we have monoliths like we have a frontend and we have a backend,” and “this will be split up into functionalities, and each functionality will be in the caption of a team.” Also, Participant BP09 approached the bounded context in a similar manner by stating “the main core business logic is in one service” and “there are different processes around our core system, and those are the ones that we split for different services.” However, the complexity increases to “first identify the correct slicing of the application and then see if there are still these cross dependencies and tackle them,” as Participant AP08 noted for establishing the size of the MSA. Soldani et al. (2018) and Taibi and Lenarduzzi (2018) identified the identification of the right size of an MSA system as concern for practitioners, which is understandable as the size of code requires an appropriate team size to maintain the software code and infrastructure resources to host the MSA system. While the boundaries are initially established, the MSA boundaries can be adjusted over the lifecycle. Participant AP05 applied an alternative strategy for cases where “it is not clear from the beginning, where we would say okay, don’t know yet how big this gets, we keep everything in one deployable unit even if we think maybe we can split this up later." Similarly, Participant BP09 approached the size problem as an active topic as "if necessary, one of my domain logics from the core service is starting to grow bigger. I can split that deployment from microservice, and that's it." While the boundaries were an essential consideration by the participants, tackling the migration of monolithic software towards MSA-based software required particular approaches.

Decomposing monolithic software and identify the bounded context can be a challenge. Eight participants referred to particular approaches to decide on the bounded context for migrating monolithic applications to MSA-based systems. Participant AP01 noted, "we have to do it step-by-step very iteratively to decouple our monoliths into the services" as a way to define the boundaries. Also, Participant BP06 recommended for the decomposition of a monolith "if you can start migrating piece by piece, right. So that's for me is the big winner on these because you can decompose the complexity piece by piece. You don't try to address it all at once." Similarly, Participant AP08 approached the migration as to "separate services, separate responsibility to the service what we were working on" to enable MSA-based systems. While Knoche and Hasselbring (2018) presented a structured approach to migrate a banking application to MSA-based system over a four year period in iterative steps that are not completed yet, the method indicated by participants seems to be driven by practical choices to deliver value quickly to the organization. Participant AP04 stated the value based decomposition approach as , "breaking down from there and to see what should be done that we can deliver" the service "and don't define it anymore by a searching into the monolith what should be answer" related to a value-driven strategy for the organization. While one aspect is the approach of decomposition of monoliths into MSA systems, another element is to enable a common understanding and language for the design of the MSA system.

A common language is required by the team to increase the understanding of the business domain and develop the MSA system based on the domain-driven design. Six participants indicated the importance of a domain-specific language in the code to improve the understanding of the business domain. While Participant AP04 noted for the design "we are working with domain-specific language," Participant AP08 suggested to "sit together with all the stakeholders involved and based on a ubiquitous language try to identify the business domains." As well, Josélyne et al. (2018) argued that MSA-based applications require a ubiquitous language for aligning with the business viewpoint. The common understanding supports the MSA team to bring "the language of the domain in our code," as stated by Participant AP05. Participant BP05 went further by saying, "there should be an alignment in the vocabulary" between the bounded context and the data model. However, cautioned at the same time to "allowing the flexibility of having different structures of vocabulary in different bounded contexts." Pautasso et al. (2017) presented the use of a ubiquitous language in the domain model as an approach for all developers working on the MSA systems to make the workflow adoptable without changing the codebase. The focus of the participants to create a shared understanding of the business perspective the use of a standard description and language eases the design and development of the MSA systems.

The establishment of the boundaries of the business domain to design the MSA systems is an iterative approach in identifications of responsibilities of the new system. The participant's focus on the identification of the boundaries aligns with the technological context of the TOE framework (see Table 1) because it supports the adoption of the MSA-based technology by making the complexity addressable by the organization. Participant AP09 explained the boundaries "helps us to shape the system, to also see, where are the dependencies and then also see the future use cases" to enable the design of the systems, which aligns well to the factors available technologies of the technology context from the TOE framework. Tornatzky and Fleischer (1990) defined the technology context with a broad definition of characteristics and availability that allows covering the focus of the participants on the enablement of the boundaries under this context.

The identification of the boundaries for MSA-based systems was of particular concern, and the findings of this subtheme tightly align with the existing literature of elicitation the bounded context of a business domain as a requirement for an MSA design (Josélyne et al., 2018). Also, Soldani et al. 2018 identified that practitioners struggle in establishing clear boundaries for the MSA design as Participant BP06 cautioned that "it's not straightforward" so that "we need to understand very well what is the challenge" and "different techniques can be applied." The identification of the business domain problem to define the boundaries was the core of the domain-driven design approach, as postulated by Thönes (2015), which is consistent with the focus of the participants. While the boundaries are crucial for MSA design, the domain-driven design does not cater for very technical concerns of the implementation.

Subtheme: Design patterns enhancing domain-driven design. In this study, 15 participants mentioned using particular design patterns to address the implementation concerns of MSA systems (see Table 4) that are not addressed by the domain-driven design approach. The most commonly used pattern is the application programming interface (API), as indicated by eight participants. An API allows to loosely couple software units and run independently by exposing their functionalities typically via a representational state transfer (REST) approach (Terzić, Dimitrieski, Kordić, Milosavljević, & Luković, 2018). The coupling between the MSA systems as per Participant BP01 was based on "restful API." While APIs simplify the connectivity between MSA systems, the approach to versioning of APIs can become challenging to address, as identified by Soldani et al. (2018). As well as Participant BP05 stated to "have a version-based API so that nothing breaks on the depended on microservice" to support a higher degree of decoupling between MSA systems. The construction of APIs is not simple and required careful construction to enable standardized communication between MSA systems by using design guidelines as mentioned by Participants AP01, AP02, and BP05. APIs have become a typical pattern applied to the MSA system to support loosely coupling as Participant BP05 stated that APIs are as "equally important concepts of domain-driven design approach when using microservice." While APIs were closely linked to MSA systems as extremely useful, the new pattern emerging as the complexity increases to address new use cases with the MSA systems.

As even loosely coupling by APIs is creating an issue for MSA systems, event-driven architecture emerged to further allow for the decoupling of MSA systems. Four participants referred to event-driven architecture to make MSA services independent from each other to ease the deployment and changes. The motivation to “decouple them with event-driven architecture” was noted by Participant AP01. Also, Participant BP04 stated, “what we do is work on event-driven architecture wherever there is a create update delete thing” that supports the decoupling form data sources of the MSA systems. Bogner, Fritzsch, Wagner, and Zimmermann (2019) identified that event-driven messaging is applied in 11 of 14 use cases for decoupling of the MSA system and asynchronous communication. Similarly, Participant BP05 referred to offload data using event-driven architecture as an approach to “replicate such data via messaging via one queue to the destination system” to decouple the MSA “in a more clearer way.” Also, Participant AP05 noted to use event-driven architecture to “only publish events, for example, and the other domain is listening on these events.” Petrasch (2017) argued that service integration requires messaging and event handling to be considered as part of the design of MSA systems. Event-driven architecture enabled participants to decouple MSA systems from each other for functions and data sources. While the emergency of event-driven architecture was increasing the use cases for MSA, other patterns exist that can be applied to MSA systems.

The use of design patterns supported the developer in addressing various challenges in the implementation of MSA systems. Ten participants mentioned using a pattern as a solution in treating different problems of the MSA system. For example, three Participants BP04, BP06, and BP08 indicated to use the strangler pattern to create an anti-corruption layer between the new and old systems. Participant BP04 was specific "to circumvent the problems of having 20 developers in one service; I went with the strangler pattern," addressing technical MSA systems issues and organizational MSA team challenges. Also, Participants AP08 and BP09 used the backend for frontend pattern to decouple the backend MSA systems from frontend applications. Participant BP04 noted, "I am also following the CQRS pattern" to enable a more performant a join of databases at the service level for data presentation over multiple MSA systems. The use of patterns allowed the participants to develop MSA services efficiently that addressed business-relevant use cases in their organizations.

Design patterns are available technology components that simplify the adoption of MSA systems for the developers. The focus of the participants on using design patterns in the design of MSA aligns with the technological context of the TOE framework. The use of the strangler patterns aligns well with factors of current equipment and methods of the technology context as Participant BP06 outlined the design of MSA system by integration with the existing systems by "composing the monolith domain by domain and you start extracting all of the code functions, creating the anti-corruption layer between the new one and the old one." Similarly, Participant BP01 stated for the event-driven architecture pattern to "integrating master data from legacy systems to offload these systems by replicating these master data to the other application," which aligns well with both factors of the technology context. The theme of design patterns tightly aligns with the technology context because the participant considered the usage of a pattern as available technology and further to integrate with current equipment and methods.

The coverage of design patterns used in the MSA area in the existing literature is limited to a few studies with a broad variety of findings. The research by Bogner et al. (2019) with an industry focus revealed the use of patterns such as event-driven, strangler, backend for frontends, and messaging for MSA-based systems, which tightly aligns with the finding of this study. However, Taibi and Lenarduzzi (2018) identified nine design patterns for the MSA-based system, which only the API pattern matched with the finding of this study because the researcher conducted a systematic mapping study and not an investigation of what pattern practitioner employed in the field. Similarly, Pahl and Jamshidi (2016) conducted a systematic mapping study, and they identified six MSA specific design patterns in which Participant BP01 similarly used only the asynchronous messaging pattern for “replicate such data via messaging via one queue.“ Also, patterns are discussed in migration scenarios that cover particular cases. For example, Furda, Fidge, Zimmermann, Kelly, and Barros (2018) presented a best practice solution for migration of legacy software code to a MSA-based software using a pattern-based migration approach for explanation only, where no pattern matched with this study because of the limited scope and complexity of the migration scenario. While MSA-based design patterns were covered by the existing literature, only design pattern investigations into practitioner-focused research align tightly with the findings of this subtheme. Practitioners tended to re-use best practices from other successful MSA-based adoptions in their environment, even when the experience was from outside their organization.

Subtheme: The importance of data management. The access to data is an essential requirement for any application in providing data to the systems' users or to other software systems. The distributed data locations of MSA-based systems increase the complexity to provide data access within the MSA system and support the access from outside the MSA system to the data as well. In this study, 11 participants agreed to the importance of data management for the design of MSA-based systems (see Table 4). As Participant AP09 noted, the question "of where certain data needs to be stored, in which system certain data should be stored" was essential for the design when working with multiple MSA-based systems. Similarly, Participant BP06 stated, "the most complex part was always around data, and that was the most challenging part always, all the time" in finding a strategy for handling the data access requirements. One of the strategies was to identify the location of the data. Participant BP04 recommended to "place the data in that hierarchy logic" but cautioned "to look at the pros and cons, while the pro is that we get this inherence automatically" and "the negative is that business logic stays in one place." Also, Participant AP04 explained, "to split up also first the business logic itself to somewhere we need to go. Then to see which data need to be transferred and then you can put on the layer of the service to see, what is my need? And in the end, this should be a big picture" of MSA services and data services. While the data transfer is one essential aspect for the location of the data, data ownership is another aspect of determining the best-suited data location.

The ownership of the data was another criterion that practitioners considered when determining the location of data in the context of MSA systems. As Participant AP05 noted regarding the multiple considerations of the data location “if the business use case also needs the service to own the data or if the service can own the data because this use case is generating the data and we don't have other services providing this data yet already?“ As Participant BP04 stated for the decision process to identify the location of the data, “I cannot give you a perfect answer. It really depends what we weigh, whether we want, complexity, not really, I mean, what type of complexity are we willing to accept?” Even the data migration process reported by Knoche and Hasselbring (2018) from a monolithic application to the MSA system has a sophisticated approach over six steps in defining various stages leading finally to an MSA system with the data set. Determine the data access pattern and data location of the MSA system was a complex decision-making process that required a lot of experience of the designer to make it well suited for the MSA-based system.

The definition and structure of the data influenced the performance of the MSA system. Six participants discussed the aspect of the data structure that enables the persistence of storage for MSA systems. Participant BP04 noted regarding the data structure to put at “least the basic attributes that are needed for the user into a database, NoSQL, or into a search engine, SQL or Elasticsearch.” Also, Participant BP06 recommended rethinking the data structure as “it's not a good idea to have a canonical data model for the whole solution. But each bounded context should come up with their own data model.” However, BP06 cautioned at the same time that “don’t make any mistakes on the datastore level or don't underestimate the importance actually and the communication between microservices themselves.” The designer required a lot of persistence layer knowledge to design an appropriate data structure to enable performance and optimal access for the MSA system. The complexity of the decision-making process for a best-suited data model and structure of an MSA system, IT managers should consider including experienced data designers into the MSA team supporting the design phase.

The design of a persistent layer required the MSA teams to adopt available technologies to support an MSA system design with an appropriate data model and data structure that was performant and maintainable. The focus of the participants on enabling the data access during the MSA adoption links to the technological context of the TOE framework because the usefulness for the software system depends on the capability to handle the data. Data designers can select the best available data structures and database management systems as Participant BP04 noted, “depending on the use case I can pick and choose the datastore,” which aligns with the technological context of the TOE framework. The focus of the participants on strategies for the ownership, location, and data management systems particularly for monolithic to MSA migration scenarios aligns well with the current equipment and methods factor to the technology context because of the importance of data assets in organizations for their business in making the right choice for the MSA adoption.

The coverage of the access to data sources in the context of MSA systems was limited in the current literature to deployment options and design patterns. Taibi and

Lenarduzzi (2018) presented data storage patterns for deployment strategies as database per service, database cluster, and shared database server pattern, which align with the Participant's BP01 statement of "our recommendation is split at database level" for the location of the data. The difficulties for establishing clear strategies for data access and data management of MSA systems by the participants were similar to the findings of Baškarada et al. (2018) because it required significant experience to understand and design distributed data management for MSA systems. Also, more straightforward migrations to MSA-based system need careful strategies for the design of the persistent layers. These strategies indicated by participants are similar to the struggles in identifying the best strategy for the data layer as reported by Balalaie et al. (2016) for migration of mobile backend service to MSA system. The challenges of participants with designing the data management seemed to be a result of existing data structures in the organization that pose the problem of ownership of the data, data locations, and data management for the design of the MSA system. Given the importance of data management in an organization, IT managers should consider including sufficient expertise for the data management design to support the appropriate decision making for the MSA-based system. The expertise and knowledge for the data management design were particularly important when the team faced migration scenarios for monolithic applications to MSA-based services that involved complex data ownership situations.

Theme 4: MSA Environment Landscape

The theme MSA environment landscape represented the technological aspects that support the lifecycle of MSA-based systems required in an organization. The focus of the participants regarding the MSA environment lead to concerns of the DevOps environment, and the platform to host MSA code. Efficiently managing the lifecycle of MSA systems are essential requirements as agile methods allowed short release cycles that pushed small code changes from the developer into production frequently using the DevOps approach. The platform that hosts the MSA-based applications is of concern not only to the team managing the MSA systems as well as to the organization to establish the right level of control and visibility for the support of the business. The strategies identified in this theme included unique strategies to address the requirements of the participant's organizational operating environment and imposed business constraints.

All 18 participants discussed strategies for the management of the MSA-based lifecycle considering customized DevOps practices and specific MSA platform configurations (see Table 5). As practitioners considered all aspects of the lifecycle for the MSA adoption (Soldani et al., 2018), I expected that participants focus on the environment required to manage and support the lifecycle of MSA. Participants indicated the importance of having a fully automated DevOps pipeline and programable platform for the MSA teams available for efficient delivery of new functions into the production environment, which Participant AP01 noted as "of course, automate as much as possible" for moving new code through the DevOps pipeline into production. Also, Participant BP01 stated to "providing transparency on the status of your infrastructure" because "everything is connected with having infrastructure as infrastructure as code" readily available for the MSA-based systems. The platform and DevOps setups were both delivered as a ready-made solution to the MSA team as Participant AP07 explained to "have an in-house Clouds running now, which we are targeting for our Microservice approach." The provisioning of such ready-made environments is a significant commitment from the organization because it requires creating an organization-specific environment addressing the organizational standards and processes as Participant AP05 referred to as "the last approval appliance, compliance step" to enable a developer to access production workloads. Providing a flexible package of the platform and DevOps increases the efficiency of the MSA developments as Participant BP07 noted: "We are deploying in our platform, we are getting everything for the DevOps" and therefore "get all the benefits" without the hassle to select and integrate tools for the development and operations of the MSA systems. The focus of the participants in having a flexible tool for the development and a proven platform available, including appropriate operational configurations, is increasing the efficiencies of the MSA systems over the lifecycle management.

Table 5

Frequency of Theme MSA Environment Landscape

	Participants		Document	
Major/Sub-theme	Count	References	Count	References
MSA Environment Landscape	18	275	5	40
DevOps to adopt MSA systems	18	206	5	27
Platform to adopt MSA systems	17	69	2	13

Note. Theme 4, MSA environment landscape; n = frequency.

The theme of the MSA environment landscape aligns well with the TOE framework because for supporting the MSA adoption, the development team must choose a development environment and an operational platform with appropriate processes and methods. MacLennan and Belle (2014) found the positive impact of standards and platforms for the SOA adoption, which aligns with the focus of the participants using a company provided pre-configured DevOps environments and platforms for the lifecycle of MSA systems. The technology context considers all relevant and available technologies that are either already in use or available in the marketplace for the adoption within the organization (Baker, 2012) which links to the intention of the organization to provide ready-made environments as Participant BP01 explained: "We have everything on the bucket from the infrastructure, the configuration, everything is a version and agnostic of the runtime environment." Also, Participant BP09 described the DevOps as "a plug and play pipeline structure" for the teams to integrate new technologies with a pre-configured DevOps pipeline to adopt MSA systems efficiently that aligns well with the technology context of the TOE framework because it allows the developer to choose and integrate appropriate technologies for the development of the MSA system. The theme MSA environment landscape was the concerns of the participants to select proper tools and methods for the MSA adoption as each MSA system required a customized environment for efficient handling during the lifecycle. Therefore, this theme MSA environment landscape aligns well with the technological context of the TOE framework for the practical adoption of the MSA system in an organization.

The researchers investigated the use of platforms and application of DevOps in the context of MSA-based systems with a focus on design studies, case studies, experience reports, and research of challenges. Larrucea et al. (2018) presented an overview of industry-grade MSA technologies with containers as an ideal deployment platform. Participant AP03 explained the use of a "microservice platform which is container-based" as a primary deployment platform for MSA-based applications in their organization. Also, Kratzke (2018) presented containers as a default platform for MSA-based applications, which aligns with the findings of this theme because of the use of containers for the MSA deployment. DevOps is becoming the typical approach to deliver MSA-based applications as Participant BP05 noted "of having an aligned DevOps toolchain" simplifies the operational support activities when appropriate skills are available. The findings of the research conducted by Baškarada et al. (2018) did not directly identify platform or DevOps concerns reported by practitioners other than challenges with skills on DevOps and containers, which emphasizes the need to address these areas proactively as similar to the findings of this theme. Pahl, Brogi, Soldani, and Jamshidi (2019) presented an overview of the container research with MSA use cases and DevOps integration for the development phase of the lifecycle, which reflected the discussion and findings of the MSA environment landscape. Also, Jamshidi et al. (2018) presented an overview of the evolution of MSA technologies, most of them were discussed by the participants and are part of this theme.

While Knoche and Hasselbring (2018) presented the concern of the platform automation and DevOps practices as part of a monolith to MSA migration case, Larrucea et al. (2018) recommended to define the platform and DevOps before the move to MSA-based applications in the organization, which aligns with the finding of the theme MSA environment landscape. Also, Bucchiarone et al. (2018) recommended to include platform automation and setting up a DevOps pipeline to increase the efficiency of the development for the MSA system. While the existing literature investigated many MSA specific topics, monolith to MSA migration case studies highlighted the need for appropriate and highly automated platform services, including a pre-defined DevOps pipeline to serve the lifecycle of MSA systems efficiently. IT managers may benefit from increased MSA system efficiency providing a set of pre-defined MSA environments that consist of platform and DevOps pipeline with a high degree of automation to the MSA teams.

Subtheme: DevOps to adopt the MSA system. All participants discussed the use of DevOps practices for the adoption of MSA systems (see Table 5). The focus by the participants regarding DevOps practices was expected as continuous integration (CI) and continuous delivery (CD) form a core of the DevOps methods. In this study, 12 participants stated to use CI for the development cycle of the MSA systems and CD for the development environments only. As Participant AP01 stated for the DevOps practices employed: “We have a continuous integration pipeline over TeamCity” and “can deploy a component directly to the open shift platform when it's there, but only in a testing environment” as not mature enough in the MSA setting to use the CD approach with external partners. Also, Participant BP04 noted to employ “continuous delivery of the environment, of a particular environment, only happens in the dev environment” to help the developer with investigations when the MSA software code was promoted into production. Regarding the CI practices, Participant BP07 explained, “we have continuous integration delivery using the Jenkins tool, and we are following, the Git flow” approach as the standard for creating feature branches and pushing the code through the DevOps pipeline. Ebert et al. (2016) recommended that each organization should customize the processes and tools towards the specifics of the MSA lifecycle requirements to enable DevOps. While the DevOps pipeline could be automated to deploy the code automatically into production, organizational processes and compliance requirements do hamper the fully automated push of code for MSA systems into production as Participant AP01 cautioned “because of the nature of our company, we have like regulations and processes to fulfill with another company. It's a little bit more complex.” While participants of Case Organization A indicated that compliance requirements impacted the automation approach of DevOps pipeline, participants of Case Organization B did not mention any substantial implications of business policies on the DevOps practices as being possibly caused by the regulatory requirements for the financial industry of Case Organization A. Also, Luz, Pinto, and Bonifácio (2019) found that practitioners struggle to identify best solutions for the constraints imposed by regulatory requirements. Contrarily, Leite et al. (2020) argued that DevOps practices both enable or hamper the ability to adhere to regulatory compliance, which aligns with the feedback of the participants. Typically, DevOps practices are supported by an adaptable and flexible set of tools to ease the handling of the various tasks leading from code development to the finally deployed packaged code.

The toolset plays a vital role in the lifecycle management of MSA-based systems as indicated by 16 participants. The clarity “on the toolchain that is going to be used for the implementation of those microservices in terms of programming language, CI/CD pipelines, libraries, frameworks, testing tools, security tools” is strongly recommended by Participant BP05 before adopting MSA in the organization. The level of integration explained by Participant AP03 that parts of toolset "integrates into our whole infrastructure and also, which is more important, I think, in our internal processes. So, we automatically create changes in the change management system" to ease the developer's effort and increase acceptance of the toolset. The number of tools in a DevOps setup can become large and can increase waste in the processes, as identified by Leite et al. (2020) and Gill et al. (2018). Participant BP08 stated, "we have Jenkins automation tools and implementing different pipelines, and we have, for instance, Sonarqube for quality" as a set of standards for the MSA-based systems. Also, Ebert et al. (2016) recommended to carefully select the right tool for enabling DevOps practices, which was indicated as necessary by 12 participants. IT managers may benefit in supporting the continuous development of the DevOps toolset with a specialized team to ensure the most suitable set of tools is considered and aligned to the MSA technologies that are made available to the organizations. While the toolset is vital to support the software engineering process through the core stages, automation of the process steps is enabling to increase speed through the pipeline.

DevOps practices required employing massive automation to deliver quickly and with a high degree of quality (Lwakatare et al., 2019). Ten participants indicated that automation is essential in providing MSA-based systems that supported the MSA team members in their lifecycle activities. Participant AP05 outlined the orchestration for the developers by "any code changes committed to the repository, triggers an automated build, all the unit tests are run, and also integration tests are run. So this is fully automated." Similarly, Participant BP06 stated, "one other thing is also important. All this continuous delivery process is fully automated. So, we basically don't depend on manuals things to execute this." However, a particular case is an automated deployment into production, which was manually staged and approved as indicated by participants (AP05, AP01) and aligns with the findings by Lwakatare et al. (2019) and Leppanen et al. (2015). IT managers will need to enable the automation of the software engineering processes and at the same time ensure that they adhere to the compliance requirements of the organization so that the MSA development team can efficiently deliver new MSA-based functionalities for the organization. Automatically deploying new software code into production requires a significant trust of the organization into the testing approach for the MSA system.

Testing is a way of confirming the readiness of the software code developed to the product owner. Seven participants referred to the testing as part of the DevOps pipeline activities. Participant AP04 noted that "test automatization is a very, very important topic" and "to check itself the structure of the code. So, the first thing is just do specific checking on code, on vulnerabilities, on syntax of easy failures." Similarly, Participant BP02 employed tests to validate the code "obviously we're using static code analysis tools." Callanan and Spillane (2016) presented a practitioner-based solution to automate the deployment validation and verification of the packages before deployed into production, which aligns with the participant's testing approach because of the missing interface testing requirement. However, Participant AP05 mentioned the challenge of the testing approach as "it's not fully automated. So, it's not like all the tests are passing, and then it's deployed into our testing environment. Also, for testing, it's a manual step."

Leite et al. (2020) argued that automating the testing is a challenging task especially when user interface tests are included which also Participant BP05 cautioned regarding the practice of end to end testing "it requires more automation in these types of environments than probably our monolithic applications." Ebert et al. (2016) recommended that a quality assurance team enables the automation of all test cases for the complete scope of code that is similar to the test strategy recommended by Participant BP06 as it "depends a lot on the strategy that we have actually, in terms of deployment. You can go for canary testing. I mean, that I would say brings some complexity as well, but at the same time, the benefit is awesome."

The testing topic was less frequently discussed by the participants as expected. Participant AP04 mentioned that testing of the MSA system was covered by the quality engineering department, and therefore is an activity outside of his MSA team. However, IT managers should be concerned with testing and enablement of full automation of the complete testing cycle for MSA systems to ensure fast return of the feedback to developers to improve the software code quality. Testing confirmed the readiness of the code to be moved into production so that the MSA systems can be monitored during operations in terms of any deviations from expected behavior.

Monitoring and alerting enable to track and react to expected and unexpected behavior of the MSA systems and trigger appropriate corrective actions if required. Six participants referred in their discussion to monitoring being an essential component to provide health information and alerting of critical events of the MSA system. Participant BP03 stated, "monitoring provided by the black box is quite simple. You just simply configure an endpoint, and it will keep calling that endpoint to check the status and report any anomaly," including "the alerting is based on a model stack, based on Prometheus alert manager and we manage everything in Opsgenie." While black-box monitoring covers the infrastructure part, white-box monitoring provides the application-centric view. Participant AP07 mentions the white box approach as "I give them tools in their hands to actually monitor the applications that they can see or that they have easy access to application logs, for instance, to see the metrics their application have generated today." Lwakatare et al. (2019) reported that monitoring of distributed systems could become a challenging task time, which was not supported by the feedback of the participants as Participant BP06 stated:

> It's much more easier to track where the problem is because you have alerts, you have monitoring, you have tracing. So, there are a lot of tools that you can use to actually identify the problem really fast, be really precise on the fix that needs to be done.

Monitoring enables the team to react to non-functional parameter deviations of the MSA systems (Leite et al., 2020), which aligned with the statement of Participant BP07 "a clear picture about the resources utilization and the load, from the user request perspective, and the number of errors and data if something goes wrong" and "we can just raise a call to our support team or create tickets in Jira tool to track things that we have worked." In this study, monitoring was of low concern of the participants because they indicated to have a very high maturity in monitoring, ensuring the availability and appropriate performance of the MSA systems deployed for the organization. However, IT managers should enable comprehensive monitoring of the MSA systems with automatic alerting for incidents that require actions by the respective team managing the system.

The subtheme DevOps to adopt MSA aligns well with the technological context of the TOE framework because of the focus of using available DevOps technologies with established customized DevOps practices to enable the efficient use of MSA systems. All participants indicated the use of DevOps practices to adopt MSA systems by building CI/CD pipelines to reduce the complexity and streamline with automation. MacLennan and Belle (2014) reported for SOA adoption that the complexity of SOA does not affect the intention to use SOA but the actual adoption of the technology which is similar with the approach taken by the participants to use DevOps practices to handle the complexity of the MSA adoption in the organization. The use of established DevOps practices enabled the organizations to introduce MSA-based systems which align well with the TOE framework.

Adopting DevOps practices for MSA-based systems is recommended by the researcher in the existing literature (Leite et al., 2020), which aligns with the focus of the participants using DevOps practices with MSA-based systems by managing the lifecycle efficiently. Also, Ebert et al. (2016) stated that MSA systems need DevOps when the focus is on efficient delivery, which aligns to Participants BP05 statement:

> I think it's important for an organization that is adopting microservices to focus first before going, let's say, full speed into a microservice implementation to be clear on the toolchain that is going to be used for the implementation of those

> microservices in terms of programming language, CI/CD pipelines, libraries, frameworks, testing tools, security tools.

Similarly, Balalaie et al. (2016) reported the success of using DevOps in a migration approach from a monolithic to MSA-based system, because it eases the coordination of development teams and reduces the deployment time for the new software code. Furthermore, Bass (2018) stated using MSA enabled the DevOps practice of Continuous Deployment (CD), which aligns with the feedback of Participant BP07. "I would say the continuous deployment is one of the most effective" practices because it allows "to deploy every single commit."

The combination of DevOps and MSA-based systems is not without challenges for the teams. Leite et al. (2020) reported several problems in using the combination of DevOps and MSA-based systems, for example, automation of integration tests, which aligns with the answers of the participants for enabling automated testing. For example, Participant AP05 detailed for testing: "it's not fully automated. So, it's not like all the tests are passing, and then it's deployed into our testing environment. Also, for testing, it's a manual step." The answers by the participants indicated that DevOps and MSA-based systems do suit each other despite the challenges caused by the MSA systems for the DevOps practices because of the flexibility in selecting tools including the automation of laborious development and operational tasks. MSA development and operations benefit from DevOps, which IT managers should consider enabling by creating a comprehensive toolchain and including sufficient DevOps expertise to improve the practice for the organization continuously.

Subtheme: Platform to adopt the MSA system. Seventeen participants indicated that MSA systems require a suitable deployment environment and support functions (see Table 5). Having a suitable platform available was anticipated as organizations tended to allow deployments of software systems that support business operations in an approved environment only. The dominant deployment approach mentioned by 11 participants was container-based deployments using the provided platform. Docker has become the de-facto standard of container technology for self-contained packaged applications such as MSA systems (Casalicchio & Iannucci, 2020). While Participant BP09 stated that "we're working with Kubernetes and Docker," Participant AP01 noted that MSA-based applications "are becoming dockerized containers in the OpenShift project." While Kubernetes is an open-source project that orchestrates container to deploy packaged MSA-based applications (Taherizadeh & Grobelnik, 2020), OpenShift is a product that is based on Kubernetes with additional functionalities and professional support (Costache, Dib, Parlavantzas, & Morin, 2017). Selecting the best-suited platform depends on the organizational decision criteria, which have not been indicated by the participants because Kubernetes and OpenShift are seen as the state-of-the-art tools for container orchestration (Casalicchio & Iannucci, 2020). In both organizations, specialized platform teams developed and provided the container-based platform to the team as noted by Participant AP06 "there is a separate platform team and the people in there, they're basically trying to come up with a technical foundation and with guidelines of how to make it such a microservice architecture possible" and Participant BP06 stated for the "So having a platform team in this case really was the big change here and in my perspective because they provide you all the scripts that you need.” Providing the platform for MSA deployments to the teams was one aspect; another aspect was the continuous improvement of the platform and alignment with the market requirements that MSA teams expected.

Ensuring a state-of-the-art platform is becoming a mandate to continuously enhancing the capabilities of the platform in supporting the MSA applications. Eight participants indicated that the continuous evolution of the platform was essential for the development of MSA-based systems because new technologies were emerging and influencing the development of MSA systems (Casalicchio & Iannucci, 2020). Participant AP03 outlined the following approach for keeping the platform up-to-date: “we don't update to the latest version immediately. We wait for two or three months, but then we do the upgrades usually.” The motivation for the alignment with the latest technology advancements is coming from the MSA teams adding new technologies into the MSA systems that require a supporting environment as Participant BP03 explained for introducing new technologies into the organization so “you can propose new technologies or new solutions, and it's evaluated, and if maybe it's approved or not.” The evolution of the platform is not in the scope of the MSA development teams instead “was provided by the platform team” that was specialized in maintaining and developing the platform services and “spread the same practices across multiple teams; so the benefit is for all and not just for one team” was noted by Participant BP06. While the platforms were provided as dedicated platform services, cloud-based platform provisioning was another way in delivering MSA-based systems.

The cloud-based delivery of MSA systems increases the flexibility and solution options for the platform team and MSA development team. Ten participants referred to the platform as a private cloud or inhouse cloud, only Participant AP07 indicated to "have an initiative to deploy to one of the big cloud vendors, AWS in our case, to bring our applications there." Both Pozdniakova and Zeika (2017) and Kratzke (2018) refer to MSA-based applications as a cloud-native application (CNA) that are best suited in cloud environments, which do not mean only public cloud environments. Moving to a public cloud vendor seems to be not the preferred option for the participant's organization because of the intense focus of the organization to deploy on ringfenced container-based platforms. The attention to develop internal capabilities of the internal cloud platform is outpaced by the innovation speed of the public cloud leaving the internal cloud quickly behind the state-of-art the MSA environment capabilities and becoming outdated.

The subtheme platform to adopt the MSA system aligns well with the TOE framework's technology context because of the focus to adopt available cloud-based technologies and build established methods to enable the deployment of MSA systems. Hsu et al. (2014) identified that the existing IT capabilities influenced the adoption of cloud computing, which aligns with the focus of the Participants regarding the use of in-house built cloud platforms for their MSA systems. Developing the platform knowledge inhouse enabled the case organizations to provide customized platform environments that are aligned to the organizational standards. Also, Oliveira, Thomas, and Espadanal (2014) presented the availability of infrastructure and IT expertise as an influencing factor for cloud adoption. Therefore, the establishment of internal capabilities and expertise around cloud-based platforms aligns well with the technological context of the TOE framework.

The use of container-based technologies for MSA-based application deployments such as Kubernetes and OpenShift is well covered by the existing literature because studies investigated the state-of-the-art container technologies (Casalicchio & Iannucci, 2020; Pahl et al., 2019). Also, the research by Baškarada et al. (2018) focused on practitioner challenges and opportunities of MSA linked DevOps and container-based technologies to MSA systems, which aligns with the finding of this theme because DevOps, Docker, and Kubernetes were mentioned by the participants as tools and platform construct. However, the results of Leite et al. (2020) identified beside the use for automated container-based deployments the use of deployment scripts, for example, Chef. In this study, no participant referred to a particular tool used for the automated deployment of MSA-based software code into containers. The existing literature reported the usage of container-based platforms for MSA systems, which aligns well with the answers of the participants in this subtheme. IT managers may want to consider building a platform team that provides an organizational aligned container-based platform for the MSA systems. The platform team should be tasked to continuously improve the platform capabilities by infusing innovation for the public cloud environment and increase the internal knowledge of MSA-based deployment approaches.

Applications to Professional Practice

This study investigated the specific IT problem, namely that some IT product managers lack strategies for the domain-driven design and DevOps practices to reduce the inefficiencies in the MSA system adoption. Participants of two case organizations provided strategies for domain-driven design and DevOps practices to optimize the efficiency in adopting MSA systems in their organizations.

The view that MSA is a genuine architectural and software technical concern of IT professionals is missing to recognize the impact of the MSA construct on the IT organization. Practitioners are concerned with the context of the MSA settings that typically include the business organization, the process environment, the technology, and the organization responsible for the lifecycle of the technology. The concern of the participants referred to the primary setting of the IT department and the alignment required to support the MSA systems efficiently. Also, Baškarada et al. (2018) identified the organizational structure and culture as a concern by practitioners for the MSA adoption. The same theme emerged recently in the literature as part of DevOps research, as Leite et al. (2020) identified the various organizational structures to adopt DevOps such as collaborating departments, cross-functional teams, and DevOps teams. The application of changing the organization structure is a change-management effort of the IT leadership outside of the scope of this study. However, the IT product manager may be able to influence the setup of the MSA team by including appropriate skills that are permanently required for the MSA lifecycle activities. Participants indicated that activities which are outside of the MSA lifecycle were not performed by the MSA team, rather performed by specialized teams such as platform teams, architecture groups, and DevOps tooling teams.

The challenges of the participants to apply the domain-driven design strategies and DevOps practices were articulated around the structure of the IT organization. While Gill et al. (2018) identified silos a boundary for interactions, Kuusinen et al. (2018) recognized that the structure of the organization limits the ability to adopt DevOps. Another aspect is that the domain-driven design elicits the business domain as a bounded context to determine the MSA size and the structure of the team. The challenges of establishing the bounded context for MSA systems, as reported by Taibi and Lenarduzzi (2018), were addressed by participants in collaborative design sessions of experienced architects, business analysts with in-depth business knowledge and other domain experts to determine the MSA boundaries. The substantial number of stakeholders for the MSA system is caused by the structure and connections to other design areas presenting a complex system (Haselböck, Weinreich, & Buchgeher, 2017). The focus of these sessions was to determine the best suited MSA size and internal complexity for independent lifecycle activities by the team. Soldani et al. (2018) reported the size and complexity of the MSA as the most dominant theme during the design stage. One participant recommended splitting the domain based on functionalities; another participant stated to divide the responsibilities of the services; another participant considered the data ownership as a factor for determining the MSA size. No clear guidance emerged, as the participants tried to balance the domain's bounded context with the technical limitations, the data management, the organizational alignment, and the independence of the teams in handling the lifecycle activities using DevOps. Another participant even recommended adjusting the boundaries over the MSA lifecycle if required, by either splitting further or combining, which required a flexible setup for the IT organization. The influence of the organization structure of business departments and IT departments on domain-design strategies and DevOps practices was salient in the participants' discussions. Therefore, enabling the alignment of the MSA team to the MSA systems and their business domain boundaries will minimize the inefficiencies over the lifecycle activities.

Experience in handling the activities and collaboration between individuals and teams was another essential aspect to address the challenges of the MSA lifecycle. The ways of working, as indicated by the participants, stressed flexible guidance in the form of expertise and best practices for design, development, and operations. Furthermore, extensive collaboration and continuous learning was another topic by the participants. Leite et al. (2020) found managerial implications to consider lean principles, training, a culture of failure, trust-building, and simplification of process adherence for DevOps. Similarly, Baškarada et al. (2018) reported the need for skilled resources, revised governance, and cross-functional operating processes as a challenge for MSA adoption. The participants agreed that extensive collaboration and lean ways of working are principles that enable DevOps and agile ways of working and suitable for MSA-based systems. One case organization used Spotify as a reference model to structure the organization and ways of working, including replicating the community meetings as a collaboration platform to improve the practices related to MSA. While changing the IT organization structure is an IT leadership effort, bringing the team together in one place to shorten the communication path was mentioned as a strategy by IT product managers as it supports agile ways of working. Following lean principles, as indicated by participants, allowed me to continuously learn and improve as an individual in gaining more experience and as an organization in identifying best practices. Also, Ebert et al. (2016) recommended close collaboration to learn and improve the ways of working. Thus, enabling active collaboration and continuous knowledge sharing will pay off in better MSA systems and efficient ways of working that allow learning and reprioritizing business needs.

Additional patterns enhance the domain-driven design for the MSA systems addressing new technology areas. The most common patterns used by the participants are restful API, strangler pattern, anti-corruption layer, event-driven architecture. The importance of patterns used for MSA systems by practitioners was reflected in a few studies only. The coverage of patterns was limited to studies from Pahl and Jamshidi (2016), Furda et al. (2018), and Taibi and Lenarduzzi (2018). Re-use is a practitioner approach to enable the replication of successful practices, in which participants indicated that the use of patterns reduced the complexity of performing MSA lifecycle activities. However, the selection and introduction of a pattern into the MSA required significant experience. Building the expertise on suitable patterns for MSA systems deployed within the organization and re-using the knowledge within the architecture group was mentioned as a possible strategy by participants.

Data management, data structure, and data location are influencing the performance, the size and the location of MSA systems. Participants struggled to present best practices for domain-driven design and DevOps, as no transparent decision approach was indicated. Soldani et al. (2018) identified the practitioner had several challenges with data storage of the MSA system for data consistency, distributed transactions, query complexity, and heterogeneity of data structures. Participants cautioned to make the decisions too quickly and not to consider multiple scenarios for selecting the best-suited approach for data management. IT product managers should infuse sufficient knowledge for the data lifecycle into the MSA team and further include the expertise of the organizational data structure during the design and development stage of the MSA systems. Identifying a suitable data management approach that supports the required performance and enables loose coupling will go a long way as revising the split of data between MSA systems at later stages can become a significant effort and burden for the MSA teams.

Providing a pre-defined MSA support environment that consists of a best practice-based DevOps pipeline and an adaptable platform to host the MSA systems is increasing the adoption efficiency. Participants considered establishing a customized DevOps pipeline for each MSA system, avoiding creating complexity by rigidly following DevOps practice. Also, Ebert et al. (2016) recommended tailoring the DevOps practices and toolchain accordingly to the best practices of the organization. As participants suggested, IT managers should consider customizing the DevOps practices and pipeline appropriately for the scope and alignment to the lifecycle requirements and not to follow the one size fits all approach. Automation is an essential principle of DevOps, agreed by participants, that increases the efficiency of the pipeline activities.

Furthermore, participants recommended for the scope of automation not only to include the pipeline activities, but also to automate the mandatory steps required by the organization. However, IT managers should observe that the toolchain is flexible enough to allow plug-and-play integrations by the MSA teams to cover specific use cases for the MSA lifecycle activities. The number of DevOps tools and required integrations can become significant, as identified by Gill et al. (2018). Providing the DevOps tooling and DevOps practices should be an effort that is located outside of the team handling the MSA systems. Participants referred to a separate DevOps team that maintained the DevOps tools and practices either as a separate team or as part of the group, providing the platform hosting the MSA systems. The responsibility split between the MSA concerns and the MSA support environment enabled them to independently improve the DevOps practices and MSA platforms aligned to the organizational requirements. Therefore, IT managers should consider building up such specialized groups acting as an enabler for MSA adoption.

Having a predefined and state-of-the-art platform increased the adoption efficiency for MSA systems. Participants used dedicated private cloud and container-based deployment platform such as Docker, Kubernetes, and OpenShift for their MSA systems. Packaging applications into containers required orchestration of multiple containers across cloud environments supporting the DevOps pipeline deployments (Pahl et al., 2019). IT managers should consider creating a set of template-based environment configurations that MSA teams can use for their deployments. The predefined components should work across cloud environments, either as a private cloud or a public cloud. While each organization will have its preferred configurations, IT managers should not mandate standardization in this area as this would increase the inefficiencies for the MSA systems. The speed of innovations in the MSA ecosystem required participants to evolve the platform to stay up to date. Thus, IT platform managers should actively seek feedback from the MSA teams regarding the features they need and collaborate on best practices for their organization to provide state-of-the-art MSA platforms.

Implications for Social Change

The adoption of MSA-based systems enforces organizations to adopt new ways of working, including introducing new supporting technologies. Typically, IT departments are organized around the plan, build and run units that may require restructuring for adopting DevOps and MSA effectively (Baškarada et al., 2018). The findings of this study may influence organization leaders to align the IT department's organizational structure to the teams that develop and operate the MSA systems. The new structure should enable the MSA teams to take full control over the lifecycle activities of the MSA systems and efficiently provide unique and innovative solutions to their stakeholders. Having a structure that allows the teams to manage the MSA independently without requesting activities by other groups increases the ability to provide value to the organization faster.

The introduction of a new structure requires clear communication and definition of responsibilities. IT managers may consider supporting the new aligned structure with change management and addressing the ways of working. A typical MSA team size is between 5 to 8 individuals who are contributors and responsible for the development and operations of a single MSA system. Providing clarity of the scope of work and responsibilities may create a better working environment for the team and individuals leading to a more efficient service delivery.

The fast innovation rate in the area of software development and MSA technologies requires individuals to acquire new knowledge regularly. Supporting an individual to learn and enhance the experience may lead to better services and personal satisfaction in delivering value to others. Increasing understanding of the individual may benefit the organization with more substantial expertise for MSA-based systems and increased value to other team members.

Agile software development methods and DevOps emphasize lean principles, continuous improvements, and open communication. IT leaders may consider enabling a culture of collaboration, learning from failures, and trust to improve the knowledge sharing of each team member. The knowledge sharing between teams and individuals in a community may improve the DevOps practices, and domain-driven design approaches for MSA systems leading to more robust software systems and sustainable business revenues, thus securing the relevance of the organization and their employees.

MSA-based systems allow building a new generation of applications providing innovative services to organizations and individuals. The continuous improvement of the practices associated with the MSA systems may enable the organizations to increase the usability of the services, including access for disadvantaged populations. The support of individuals in their daily life by removing barriers and increasing the accessibility to the services offered may create a positive impact for the community.

Recommendations for Action

Applying strategies of domain-driven design and DevOps practices for the efficient adoption of MSA-based systems, practitioners will need to consider the specifics of the organizational setup, establish a culture of collaboration, allow to learn and improve, and provide an MSA suitable environment. The first recommendation for leaders of IT organizations is that to adopt an IT organization structure aligned with the business context of the MSA system. The alignment of the team to the business context of the MSA allows employing a full lifecycle approach enabling to fulfill the business demand efficiently. The definition of the boundaries of MSA should be identified by experienced domain-driven design practitioners so that a highly independent operating team can be established for the lifecycle activities. Furthermore, the DevOps principle ‘we build it - we run it’ should not be hampered by silos in the IT department, establishing the required skills and capabilities within the team. The teams should be equipped with appropriate expertise to manage the design, development, and operations independently. While the focus is on supporting the business with MSA-based systems, the teams will need to have an appropriate support environment available for all other topics outside of the MSA system context. The concerns of the teams can relate to the MSA hosting platforms, architecture patterns and decisions, best practices for DevOps, compliance requirements, and the application of appropriate security. Without a supporting structure, the MSA teams would deviate the focus away from actioning the tasks of the MSA systems lifecycle activities.

The second recommendation for the IT manager is to establish a culture of knowledge being shared in communities, active collaboration, and continuous learning and improving. Agile ways of working need guiding principles, free flow of information, short communication lines, and lean methods for focusing on the priorities of business requests. Thus, providing on-demand-based guidance for the teams in form of architecture expertise, best practice handbooks, automation of compliance steps, and community-based knowledge should allow to simplify the decision-making process with proven organizational assets and also encouraging the participation in community sessions to improve the practice by fostering the collaboration across MSA teams similar to open-source driven principles. IT managers should encourage active knowledge acquisition at an individual level by providing training or support the participation at practitioner events to increase the level of expertise within the teams. As an MSA team should cover the complete lifecycle activities independently, strong expertise and quick access to best practice-based knowledge are essential to adopt MSA systems efficiently.

The third recommendation for IT product managers is to support the establishment of the domain boundaries for the MSA systems based on the shared understanding of the MSA teams, the technical implementation patterns, and the required access to the data structure of the organization. The business boundaries are not always clearly defined, which leads to difficulties in the identification and appropriate MSA slicing. Enabling the team in gaining knowledge and bringing in focused business expertise mitigates better cross dependencies to other domains supporting the independence of the MSA system. While domain-driven-design creates a strong business domain focus, the technical implementation for MSA systems can become complicated. IT managers should allow appropriate functional implementation patterns such as strangler, anti-corruption layer, backend-for-frontend, and others that simplify the MSA deployment and increasing the efficiency of the MSA system. Furthermore, IT managers should observe the access to data requirements in order to avoid a dependency on other software systems by balancing the data structure requirements, data location, and data ownership, not leading to a constraint for the MSA lifecycle activities.

The fourth recommendation for IT managers is to provide an organizational contextualized MSA environment to the MSA teams in form of a customized plug and play DevOps pipeline based on DevOps practices and ready-made MSA platforms for development and production. IT managers should establish a DevOps pipeline that automates not only the DevOps best practice activities but also any other compliance requirements of the organization for each phase of the lifecycle. The DevOps pipeline should be freely configurable with as few mandatory DevOps components as possible to support the team in their MSA lifecycle stage. Furthermore, to complete the MSA environment, the IT manager should provide a safe and secure environment to host the MSA systems. This environment should have all components ready such as monitoring, versioning support, deployment scripts, logging, dashboards, and others that might be necessary for the deployment of the MSA systems via the DevOps pipeline. While providing a set of predefined MSA environment components increases the re-use and efficiency, this should not become a mandatory need for the MSA teams avoiding increased inefficiencies of other requirements.

In addition to IT managers engaged with MSA systems, this study might be relevant for enterprise IT leaders who want to align to the business organization better and who are challenged by the complexity of adopting MSA-based software systems. I might disseminate the finding of this study in appropriate formats via events, training, conferences, and as part of my professional work. Additionally, I will distribute a copy of this research to all gatekeepers, participants, and individuals who helped me to get connected to possible case organizations.

Recommendations for Further Study

The findings of this research and the related assumptions, limitations, and delimitations guide further research in the area of MSA. This qualitative multiple-case study investigated two large global operating case organizations, one in the customer goods manufacturing industry and one in the banking industry with headquarters in Europe. The first recommendation is to compare the finding by exploring case organizations of other industries using a qualitative approach with a similar design. In this study, the case organizations were operating under various country-specific regulations depending on the industry and adopting ways to comply with these for MSA systems. The compliance drives particular ways of working and organizational support structure. Understanding possible differences to other industries and geography might help others to apply the strategies to the readers' organization setting and environment.

While the organizational and geographical context might provide some insights, the involved participants used these strategies to the MSA environment.

The participants of this study belonged to decision-makers related to parts of the lifecycle of MSA systems from analysis, design, development to operations, and platforms. Larger organizations tend to distribute analysis, design, development, operations, and infrastructure concerns of MSA systems to various teams and departments as part of their approach and organization structure. The themes identified are a direct reflection of the involvement of participants that come from various functional teams. Another recommendation is to investigate using a more cohesive participant group or IT executives with concerns related to MSA systems. This research could help to elicit further details and enhance the understanding of the strategies applied for MSA systems. The application of strategies are supporting practitioners to adopt MSA systems that enable particular objectives of the organization and protecting the value in legacy systems.

Enterprise organizations rarely develop a new system without dependencies on legacy systems and involvement of an internal support organization. One particular discovery of this study was the challenge caused by legacy systems that exist in the organization for the adoption of MSA-based systems. The problems related to the extraction, development, operations of MSA systems, and the organizational support structures for the adoption. Researchers could investigate the migration of MSA systems so that strategies can be identified that support the design, development and operations during the migration phase. Migration research would be of value to practitioners in order to improve existing application systems for their users by moving towards MSA systems. Another recommendation would be to research the challenges of existing organizational structures impacting the MSA systems development. This study identified, the development of MSA systems is affected by the bounded context that exists for the domain and the organization supporting these domains. Therefore, I also would recommend research of the organizational aspects related to the MSA systems with a focus on the IT organization. Gaining more profound insights into the relationship between the development of MSA systems and the exiting supporting organization would help to develop particular strategies easing the adoption.

Reflections

Starting this journey of pursuing a doctoral degree was an easy decision without realizing the significant effort needed by myself and the burden I placed on my family. While at the beginning of the research program, it was easy to juggle between family, work, and pursuit, towards the end, it became a significant impact on my family in supporting me during the doctoral program. Encouraging support from my family allowed me to work over weekends on my research to complete the program. As a non-native English speaker, I have been writing business documents in the English language for a couple of years. However, writing in a scholarly form took me some time to adapt.

The investigation into a research topic and the discussion in an appropriate academic format transformed my everyday professional communication much more than I expected. As an IT practitioner for multiple decades, I have always been interested in the impact that particular architecture software systems have on enterprise organizations.

However, the focus of my work is based on large scale infrastructure outsourcing projects with limited exposure to software development. The noise in the industry that MSA created and the possible impact caused in the ways of working motivated me to investigate further and enhance my knowledge on this topic.

The acquisition of case organizations proved to be difficult for my research study, despite many organizations publishing and presenting their MSA adoption progress at events. My eagerness to conduct interviews may have introduced some bias towards the selection and acceptance of suitable case organizations. I attempted to mitigate any issues with the quality and number of possible participants by focusing on large and globally operating organizations. While I tried to be as careful as possible to not introduce bias during the data collection and analysis, I could have unintentionally or unknowingly influenced the interviews and the analysis of the data as having in-depth industry knowledge. I possibly mitigated the misrepresentation of the data by the direct traceability of the findings to the evidence. My learning from this research is that past organizational and current organizational structures affect the design, development, and operations of MSA-based software systems.

Summary and Study Conclusions

The adoption of MSA systems into an organization is a complex undertaking. The tasks for IT leaders and managers involved adopting new ways of working for the lifecycle activities of MSA-based systems. MSA-based systems required an aligned IT organization structure including a culture of learning and improving for the engaged individuals. The building of new knowledge was supported by providing of pre-configured capabilities and infusing of external expertise to develop a set of best practices. While agile software engineering practices influence the principles and initial set of methods for domain-driven design and DevOps, the next level of detailing these practices required significant experience and customizations to the organizational context and MSA scope. MSA-based applications allowed to support changing business requirements with fast delivery methods but demand a high trust and collaborative IT organization with an independent setup for the lifecycle management.

Domain-driven design is supporting the determination of the business domain's bounded context; however, the MSA system may need additional support through design patterns to enable the technical implementation. Also, during the design phase, significant experience of the designers and architects were required to address the complexity of the business domain, data management, and technical design decisions. While MSA paired well with DevOps, establishing a customized tooling chain and a set of pre-defined practices supports the efficiency of the MSA adoption. The deployment environment for the MSA system should be tight to DevOps practices using state-of-the-art container platforms with extensive automation to cover the complete DevOps pipeline. The ecosystem of DevOps and container platforms need continuous refinement and infusion of innovations to stay relevant for the MSA deployment environment.

References

Abdalla, M. M., Oliveira, L. G. L., Azevedo, C. E. F., & Gonzalez, R. K. (2018). Quality in qualitative organizational research: Types of triangulation as a methodological alternative. *Administração: Ensino e Pesquisa*, *19*(1), 66–98. doi:10.13058/raep.2018.v19n1.578

Ahmadi, H., Nilashi, M., & Ibrahim, O. (2015). Organizational decision to adopt hospital information system: An empirical investigation in the case of Malaysian public hospitals. *International Journal of Medical Informatics*, *84*(3), 166–188. doi:10.1016/j.ijmedinf.2014.12.004

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, *50*(2), 179–211. doi:10.1016/0749-5978(91)90020-T

Al-Hujran, O., Al-Lozi, E. M., Al-Debei, M. M., & Maqableh, M. (2018). Challenges of cloud computing adoption from the TOE framework perspective. *International Journal of E-Business Research*, *14*(3), 77–94. doi:10.4018/IJEBR.2018070105

Alshenqeeti, H. (2014). Interviewing as a data collection method: A critical review. *English Linguistics Research*, *3*(1), 233–235. doi:10.5430/elr.v3n1p39

Alshuqayran, N., Ali, N., & Evans, R. (2016). A systematic mapping study in microservice architecture. In *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)* (pp. 44–51). IEEE. doi:10.1109/SOCA.2016.15

Amaral, M., Polo, J., Carrera, D., Mohomed, I., Unuvar, M., & Steinder, M. (2015). Performance evaluation of microservices architectures using containers. In *2015*

IEEE 14th International Symposium on Network Computing and Applications (pp. 27–34). IEEE. doi:10.1109/NCA.2015.49

Annosi, M. C., Magnusson, M., Martini, A., & Appio, F. P. (2016). Social Conduct, Learning and Innovation: An Abductive Study of the Dark Side of Agile Software Development. *Creativity and Innovation Management*, *25*(4), 515–535. doi:10.1111/caim.12172

Arsel, Z. (2017). Asking questions with reflexive focus: A tutorial on designing and conducting interviews. *Journal of Consumer Research*, *44*(4), 939–949. doi:10.1093/jcr/ucx096

Atkin, D. J., Hunt, D. S., & Lin, C. A. (2015). Diffusion theory in the new media environment: toward an integrated technology adoption model. *Mass Communication and Society*, *18*(5), 623–650. doi:10.1080/15205436.2015.1066014

Awa, H. O., Ojiabo, O. U., & Orokor, L. E. (2017a). Integrated technology-organization-environment (T-O-E) taxonomies for technology adoption. *Journal of Enterprise Information Management*, *30*(6), 893–921. doi:10.1108/JEIM-03-2016-0079

Awa, H. O., Ukoha, O., & Igwe, S. R. (2017b). Revisiting technology-organization-environment (T-O-E) theory for enriched applicability. *Bottom Line*, *30*(1), 2–22. doi:10.1108/BL-12-2016-0044

Baker, J. (2012). The Technology–Organization–Environment Framework. In Y. K. Dwivedi, M. R. Wade, & S. L. Schneberger (Eds.) (Vol. 29, pp. 231–245). New York, NY: Springer. doi:10.1007/978-1-4419-6108-2_12

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables DevOps: Migration to a cloud-native architecture. *IEEE Software*, *33*(3), 42–52. doi:10.1109/MS.2016.64

Bansal, P. T., Smith, W. K., & Vaara, E. (2018). New ways of seeing through qualitative research. *Academy of Management Journal*, *61*(4), 1189–1195. doi:10.5465/amj.2018.4004

Baškarada, S. (2014). Qualitative Case Study Guidelines. *Qualitative Report*, *19*(40), 1–25. doi:10.7748/nr2013.05.20.5.28.e327

Baškarada, S., Nguyen, V., & Koronios, A. (2018). Architecting microservices: Practical opportunities and challenges. *Journal of Computer Information Systems*, 1–9. doi:10.1080/08874417.2018.1520056

Bass, L. (2018). The software architect and DevOps. *IEEE Software*, *35*(1), 8–10. doi:10.1109/MS.2017.4541051

Batubara, F. R., Ubacht, J., & Janssen, M. (2018). Challenges of blockchain technology adoption for e-government. In *Proceedings of the 19th Annual International Conference on Digital Government Research Governance in the Data Age - dgo '18* (pp. 1–9). New York, NY: ACM Press. doi:10.1145/3209281.3209317

Benoot, C., Hannes, K., & Bilsen, J. (2016). The use of purposeful sampling in qualitative evidence synthesis: A worked example on sexual adjustment to a cancer trajectory. *BMC Medical Research Methodology*, *16*(1), 1–13. doi:10.1186/s12874-016-0114-6

Bhattacharya, M. (2015). A conceptual framework of RFID adoption in retail using

Rogers stage model. *Business Process Management Journal*, *21*(3), 517–540. doi:10.1108/BPMJ-06-2014-0047

Biggs, J. S. G., & Marchesi, A. (2015). Information for consent: Too long and too hard to read. *Research Ethics*, *11*(3), 133–141. doi:10.1177/1747016115583381

Billo, E., & Mountz, A. (2016). For institutional ethnography: Geographical approaches to institutions and the everyday. *Progress in Human Geography*, *40*(2), 199–220. doi:10.1177/0309132515572269

Birt, L., Scott, S., Cavers, D., Campbell, C., & Walter, F. (2016). Member checking: A tool to enhance trustworthiness or merely a nod to validation? *Qualitative Health Research*, *26*(13), 1802–1811. doi:10.1177/1049732316654870

Blair, E. (2015). A reflexive exploration of two qualitative data coding techniques. *Journal of Methods and Measurement in the Social Sciences*, *6*(1), 14. doi:10.2458/v6i1.18772

Blair, E., & Deacon, A. (2015). A holistic approach to fieldwork through balanced, reflective practice. *Reflective Practice*, *16*(3), 418–434. doi:10.1080/14623943.2015.1052388

Blind, K., Petersen, S. S., & Riillo, C. A. F. (2017). The impact of standards and regulation on innovation in uncertain markets. *Research Policy*, *46*(1), 249–264. doi:10.1016/j.respol.2016.11.003

Boeren, E. (2018). The methodological underdog: A review of quantitative research in the key adult education journals. *Adult Education Quarterly*, *68*(1), 63–79. doi:10.1177/0741713617739347

Bogner, J., Fritzsch, J., Wagner, S., & Zimmermann, A. (2019). Assuring the Evolvability of Microservices: Insights into Industry Practices and Challenges. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 546–556). IEEE. doi:10.1109/ICSME.2019.00089

Bogner, J., & Zimmermann, A. (2016). Towards integrating microservices with adaptable enterprise architecture. In *2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW)* (pp. 1–6). IEEE. doi:10.1109/EDOCW.2016.7584392

Borrett, D. S., Sampson, H., & Cavoukian, A. (2017). Research ethics by design: A collaborative research design proposal. *Research Ethics*, *13*(2), 84–91. doi:10.1177/1747016116673135

Bossle, M. B., Dutra De Barcellos, M., Vieira, L. M., & Sauvée, L. (2016). The drivers for adoption of eco-innovation. *Journal of Cleaner Production*, *113*, 861–872. doi:10.1016/j.jclepro.2015.11.033

Bradford, M., Earp, J. B., & Grabski, S. (2014). Centralized end-to-end identity and access management and ERP systems: A multi-case analysis using the technology organization environment framework. *International Journal of Accounting Information Systems*, *15*(2), 149–165. doi:10.1016/j.accinf.2014.01.003

Braun, V., & Clarke, V. (2016). (Mis)conceptualizing themes, thematic analysis, and other problems with Fugard and Potts' (2015) sample-size tool for thematic analysis. *International Journal of Social Research Methodology*, *19*(6), 739–743. doi:10.1080/13645579.2016.1195588

Bruce, A., Beuthin, R., Sheilds, L., Molzahn, A., & Schick-Makaroff, K. (2016). Narrative research is evolving: Evolving through narrative research. *International Journal of Qualitative Methods*, *15*(1), 1–6. doi:10.1177/1609406916659292

Bucchiarone, A., Dragoni, N., Dustdar, S., Larsen, S. T., & Mazzara, M. (2018). From monolithic to microservices: An experience report from the banking domain. *IEEE Software*, *35*(3), 50–55. doi:10.1109/MS.2018.2141026

Butler, C. (2015). Making interview transcripts real: the reader's response. *Work, Employment and Society*, *29*(1), 166–176. doi:10.1177/0950017014523482

Callanan, M., & Spillane, A. (2016). DevOps: Making it easy to do the right thing. *IEEE Software*, *33*(3), 53–59. doi:10.1109/MS.2016.66

Carmel-Gilfilen, C., & Portillo, M. (2016). Designing with empathy: Humanizing narratives for inspired healthcare experiences. *Health Environments Research and Design Journal*, *9*(2), 130–146. doi:10.1177/1937586715592633

Carolan, C. M., Forbat, L., & Smith, A. (2016). Developing the DESCARTE model. *Qualitative Health Research*, *26*(5), 626–639. doi:10.1177/1049732315602488

Casalicchio, E., & Iannucci, S. (2020). The state-of-the-art in container technologies: Application, orchestration and security. *Concurrency and Computation: Practice and Experience*. doi:10.1002/cpe.5668

Castillo-Montoya, M. (2016). Preparing for interview research : The interview protocol refinement framework. *The Qualitative Report*, *21*(5), 811–831. Retrieved from http://nsuworks.nova.edu/

Castleberry, A., & Nolen, A. (2018). Thematic analysis of qualitative research data: Is it as easy as it sounds? *Currents in Pharmacy Teaching and Learning*, *10*(6), 807–815. doi:10.1016/j.cptl.2018.03.019

Cerny, T., Donahoo, M. J., & Trnka, M. (2018). Contextual understanding of microservice architecture. *ACM SIGAPP Applied Computing Review*, *17*(4), 29–45. doi:10.1145/3183628.3183631

Chan, N. N., Walker, C., & Gleaves, A. (2015). An exploration of students' lived experiences of using smartphones in diverse learning contexts using a hermeneutic phenomenological approach. *Computers and Education, 82*, 96–106. doi:10.1016/j.compedu.2014.11.001

Chandra, S., & Kumar, K. N. (2018). Exploring factors influencing organizational adoption of augmented reality in e-commerce: Empirical analysis using technology-organization-environment model. *Journal of Electronic Commerce Research*, *19*(3), 237–265. Retrieved from http://www.jecr.org/

Chase, E. (2017). Enhanced member checks: Reflections and insights from a participant-researcher collaboration. *The Qualitative Report*, *22*(10), 2689–2703. Retrieved from http://nsuworks.nova.edu/tqr/

Chowdhury, M. F. (2015). Coding, sorting and sifting of qualitative data analysis: debates and discussion. *Quality and Quantity*, *49*(3), 1135–1143. doi:10.1007/s11135-014-0039-2

Claps, G. G., Berntsson Svensson, R., & Aurum, A. (2015). On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software Technology*, *57*(1), 21–31. doi:10.1016/j.infsof.2014.07.009

Clarke, V., & Braun, V. (2017). Thematic analysis. *Journal of Positive Psychology*, *12*(3), 297–298. doi:10.1007/BF00791193

Connelly, L. M. (2016). Trustworthiness in qualitative research. *MEDSURG Nursing*, *25*(6), 435–436. Retrieved from https://library.amsn.org/amsn/

Constantinou, C. S., Georgiou, M., & Perdikogianni, M. (2017). A comparative method for themes saturation (CoMeTS) in qualitative interviews. *Qualitative Research*, *17*(5), 571–588. doi:10.1177/1468794116686650

Cope, D. G. (2014). Computer-assisted qualitative data analysis software. *Oncology Nursing Forum*, *41*(3), 322–323. doi:10.1188/14.ONF.322-323

Costache, S., Dib, D., Parlavantzas, N., & Morin, C. (2017). Resource management in cloud platform as a service systems: Analysis and opportunities. *Journal of Systems and Software*, *132*, 98–118. doi:10.1016/j.jss.2017.05.035

Crowe, M., Inder, M., & Porter, R. (2015). Conducting qualitative research in mental health: Thematic and content analyses. *Australian & New Zealand Journal of Psychiatry*, *49*(7), 616–623. doi:10.1177/0004867415582053

Cuesta, C. E., Navarro, E., & Zdun, U. (2016). Synergies of system-of-systems and microservices architectures. In *Proceedings of the International Colloquium on Software-intensive Systems-of-Systems at 10th European Conference on Software Architecture - SiSoS@ECSA '16* (Vol. 3, pp. 1–7). New York, USA: ACM Press. doi:10.1145/3175731.3176176

Daniel, N. E., & Cooper, P. B. (2017). The relationship between personal values, organizational formalization and employee work outcomes of compliance and innovation. *International Journal of Manpower*, *38*(2), 274–287. doi:10.1108/IJM-06-2015-0090

Danylenko, Y. A. (2018). Characteristics and classification of innovation and innovation process. *Science and Innovation*, *14*(3), 14–26. doi:10.15407/scine14.03.014

Dasgupta, M. (2015). Exploring the relevance of case study research. *Vision: The Journal of Business Perspective*, *19*(2), 147–160. doi:10.1177/0972262915575661

Davis, F. (1986). *A technology acceptance model for empirically testing new end-use information systems: Theory and results*. Massachusetts Institute of Technology.

Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319. doi:10.2307/249008

Deakin, H., & Wakefield, K. (2014). Skype interviewing: Reflections of two PhD researchers. *Qualitative Research*, *14*(5), 603–616. doi:10.1177/1468794113488126

Dekoulou, P., & Trivellas, P. (2017). Organizational structure, innovation performance, and customer relationship value in the Greek advertising and media industry. *Journal of Business & Industrial Marketing*, *32*(3), 385–397. doi:10.1108/JBIM-07-2015-0135

De Massis, A., & Kotlar, J. (2014). The case study method in family business research: Guidelines for qualitative scholarship. *Journal of Family Business Strategy*, *5*(1), 15–29. doi:10.1016/j.jfbs.2014.01.007

Dempsey, L., Dowling, M., Larkin, P., & Murphy, K. (2016). Sensitive interviewing in qualitative research. *Research in Nursing and Health*, *39*(6), 480–490.

doi:10.1002/nur.21743

Dennis, B. (2018). Validity as research praxis: A study of self-reflection and engagement in qualitative inquiry. *Qualitative Inquiry*, *24*(2), 109–118. doi:10.1177/1077800416686371

Denscombe, M. (2013). The role of research proposals in business and management education. *The International Journal of Management Education*, *11*(3), 142–149. doi:10.1016/j.ijme.2013.03.001

Deterding, N. M., & Waters, M. C. (2018). Flexible coding of in-depth interviews. *Sociological Methods & Research*, 1–32. doi:10.1177/0049124118799377

Dikko, M. (2016). Establishing construct validity and reliability: Pilot testing of a qualitative interview for research in Takaful (Islamic Insurance). *Qualitative Report*, *21*(3), 521–528. Retrieved from http://nsuworks.nova.edu/tqr

Doody, O., & Noonan, M. (2016). Nursing research ethics, guidance, and application in practice. *British Journal of Nursing*, *25*(14), 803–807. doi:10.12968/bjon.2016.25.14.803

Dooley, L., Kenny, B., & Cronin, M. (2016). Interorganizational innovation across geographic and cognitive boundaries: Does firm size matter? *R and D Management*, *46*, 227–243. doi:10.1111/radm.12134

Draper, J. (2015). Ethnography: Principles, practice, and potential. *Nursing Standard*, *29*(36), 36–41. doi:10.7748/ns.29.36.36.e8937

Drouin, M., Stewart, J., & Van Gorder, K. (2015). Using methodological triangulation to examine the effectiveness of a mentoring program for online instructors. *Distance*

Education, *36*(3), 400–418. doi:10.1080/01587919.2015.1081735

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software*, *33*(3), 94–100. doi:10.1109/MS.2016.68

Ehteshami, A. (2017). Barcode technology acceptance and utilization in health information management department at academic hospitals according to technology acceptance model. *Acta Informatica Medica*, *25*(1), 4. doi:10.5455/aim.2017.25.4-8

Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*, *29*(6), 1–21. doi:10.1002/smr.1885

Evans, E. (2004). *Domain-driven design: Tackling complexity in the heart of software*. Boston, MA: Pearson Education

Evwiekpaefe, A. E., Chiemeke, S. C., & Haruna, M. Z. (2018). Individual and organizational acceptance of technology theories and models : Conceptual gap and possible solutions. *Pacific Journal of Science and Technology*, *19*(1), 189–197. Retrieved from http://www.akamaiuniversity.us/PJST.htm

Fetzer, C. (2016). Building critical applications using microservices. *IEEE Security & Privacy*, *14*(6), 86–89. doi:10.1109/MSP.2016.129

FitzPatrick, B. (2019). Validity in qualitative health education research. *Currents in Pharmacy Teaching and Learning*, *11*(2), 211-217. doi:10.1016/j.cptl.2018.11.014

Foe, G., & Larson, E. L. (2016). Reading level and comprehension of research consent forms. *Journal of Empirical Research on Human Research Ethics*, *11*(1), 31–46. doi:10.1177/1556264616637483

Folkinshteyn, D., & Lennon, M. (2016). Braving bitcoin: A technology acceptance model (TAM) analysis. *Journal of Information Technology Case and Application Research*, *18*(4), 220–249. doi:10.1080/15228053.2016.1275242

Forero, R., Nahidi, S., De Costa, J., Mohsin, M., Fitzgerald, G., Gibson, N., … Aboagye-Sarfo, P. (2018). Application of four-dimension criteria to assess rigour of qualitative research in emergency medicine. *BMC Health Services Research*, *18*(1), 1–11. doi:10.1186/s12913-018-2915-2

Forés, B., & Camisón, C. (2016). Does incremental and radical innovation performance depend on different types of knowledge accumulation capabilities and organizational size? *Journal of Business Research*, *69*(2), 831–848. doi:10.1016/j.jbusres.2015.07.006

Foss, N. J., & Hallberg, N. L. (2014). How symmetrical assumptions advance strategic management research. *Strategic Management Journal*, *35*(6), 903–913. doi:10.1002/smj.2130

Francesco, P. D., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*, *150*(4), 77–97. doi:10.1016/j.jss.2019.01.001

Francesco, P. D., Malavolta, I., & Lago, P. (2017). Research on architecting Microservices: Trends, focus, and potential for industrial adoption. In *2017 IEEE International Conference on Software Architecture (ICSA)* (pp. 21–30). IEEE.

doi:10.1109/ICSA.2017.24

Friesen, P., Kearns, L., Redman, B., & Caplan, A. L. (2017). Rethinking the Belmont report? *The American Journal of Bioethics*, *17*(7), 15–21. doi:10.1080/15265161.2017.1329482

Fritzsch, J., Bogner, J., Wagner, S., & Zimmermann, A. (2019). Microservices Migration in Industry: Intentions, Strategies, and Challenges. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 481–490). IEEE. doi:10.1109/ICSME.2019.00081

Fuller, R. (2019). Functional Organization of Software Groups Considered Harmful. In *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)* (pp. 120–124). IEEE. doi:10.1109/ICSSP.2019.00024

Furda, A., Fidge, C., Zimmermann, O., Kelly, W., & Barros, A. (2018). Migrating enterprise legacy source code to microservices: On multitenancy, statefulness, and data consistency. *IEEE Software*, *35*(3), 63–72. doi:10.1109/MS.2017.440134612

Fusch, P. I., & Ness, L. R. (2015). Are we there yet? Data saturation in qualitative research. *The Qualitative Report*, *20*(9), 1408–1416. Retrieved from http://tqr.nova.edu/wp-content/uploads/2015/09/

Gabbrielli, M., Giallorenzo, S., Guidi, C., Mauro, J., & Montesi, F. (2016). Self-reconfiguring microservices. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9660, pp. 194–210). doi:10.1007/978-3-319-30734-3_14

Gagnon, M., Jacob, J. D., & McCabe, J. (2015). Locating the qualitative interview:

reflecting on space and place in nursing research. *Journal of Research in Nursing*, *20*(3), 203–215. doi:10.1177/1744987114536571

Ganapathy, M. (2016). Qualitative data analysis: Making it easy for the nurse researcher. *International Journal of Nursing Education*, *8*(2), 106. doi:10.5958/0974-9357.2016.00057.X

Gangwar, H., Date, H., & Raoot, A. D. (2014). Review on IT adoption: Insights from recent technologies. *Journal of Enterprise Information Management*, *27*(4), 488–502. doi:10.1108/JEIM-08-2012-0047

Garbarski, D., Schaeffer, N. C., & Dykema, J. (2016). Interviewing practices, conversational practices, and rapport: Responsiveness and engagement in the standardized survey interview. *Sociological Methodology*, *46*(1), 1–38. doi:10.1177/0081175016637890

Gemünden, H. G., Lehner, P., & Kock, A. (2018). The project-oriented organization and its contribution to innovation. *International Journal of Project Management*, *36*(1), 147–160. doi:10.1016/j.ijproman.2017.07.009

Gentles, S. J., Charles, C., & Ploeg, J. (2015). Sampling in qualitative research: Insights from an overview of the methods literature. *The Qualitative Report*, *20*(11), 1772–1789. doi:10.4135/9781412950589.n885

Gergen, K. J., Josselson, R., & Freeman, M. (2015). The promises of qualitative inquiry. *American Psychologist*, *70*(1), 1–9. doi:10.1037/a0038597

Gill, A. Q., Loumish, A., Riyat, I., & Han, S. (2018). DevOps for information management systems. *VINE Journal of Information and Knowledge Management*

Systems, *48*(1), 122–139. doi:10.1108/VJIKMS-02-2017-0007

Gill, M. J. (2014). The possibilities of phenomenology for organizational research. *Organizational Research Methods*, *17*(2), 118–137. doi:10.1177/1094428113518348

Gottinger, H. W. (2016). Innovation, dynamics of competition and market dynamics. *Archieves of Business Research*, *4*(1), 1–16. doi:10.14738/abr.41.1737

Grossoehme, D. H. (2014). Overview of qualitative research. *Journal of Health Care Chaplaincy*, *20*(3), 109–122. doi:10.1080/08854726.2014.925660

Grysman, A., & Mansfield, C. D. (2017). What do we have when we have a narrative? *Imagination, Cognition, and Personality: Consciousness in Theory, Research, and Clinical Practice*, *37*(2), 105–126. doi:10.1177/0276236617733823

Hadi, M. A., & Closs, S. J. (2016). Ensuring rigour and trustworthiness of qualitative research in clinical pharmacy. *International Journal of Clinical Pharmacy*, *38*(3), 641–646. doi:10.1007/s11096-015-0237-6

Hagaman, A. K., & Wutich, A. (2017). How many interviews are enough to identify meta-themes in multisited and cross-cultural research? Another perspective on guest, gunce, and johnson's (2006) landmark study. *Field Methods*, *29*(1), 23–41. doi:10.1177/1525822X16640447

Hammersley, M. (2017). Interview data: a qualified defense against the radical critique. *Qualitative Research*, *17*(2), 173–186. doi:10.1177/1468794116671988

Haselböck, S., Weinreich, R., & Buchgeher, G. (2017). Decision Models for Microservices: Design Areas, Stakeholders, Use Cases, and Requirements. In

Architecture (Vol. 1, pp. 155–170). doi:10.1007/978-3-319-65831-5_11

Hashmi, A. R., & Van Biesebroeck, J. (2016). The relationship between market structure and innovation in industry equilibrium: A case study of the global automobile industry. *Review of Economics and Statistics*, *98*(1), 192–208. doi:10.1162/REST_a_00494

Hassan, S., & Bahsoon, R. (2016). Microservices and their design trade-offs: A self-adaptive roadmap. In *2016 IEEE International Conference on Services Computing (SCC)* (pp. 813–818). IEEE. doi:10.1109/SCC.2016.113

Hasselbring, W., & Steinacker, G. (2017). Microservice architectures for scalability, agility and reliability in E-Commerce. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)* (pp. 243–246). IEEE. doi:10.1109/ICSAW.2017.11

Hays, D. G., Wood, C., Dahl, H., & Kirk-Jenkins, A. (2016). Methodological rigor in journal of counseling & development qualitative research articles: A 15-year review. *Journal of Counseling and Development*, *94*(2), 172–183. doi:10.1002/jcad.12074

Hazzan, O., & Nutov, L. (2014). Teaching and learning qualitative research ≈ conducting qualitative research. *The Qualitative Report*, *19*(24), 1–29. Retrieved from http://nsuworks.nova.edu/tqr/vol19/iss24/3/

He, Z. (2015). Rivalry, market structure and innovation: The case of mobile banking. *Review of Industrial Organization*, *47*(2), 219–242. doi:10.1007/s11151-015-9466-z

Hekkala, R., Stein, M.-K., Rossi, M., & Smolander, K. (2017). Challenges in Transitioning to an Agile Way of Working. *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*, 5869–5878. doi:10.24251/hicss.2017.707

Hennink, M. M., Kaiser, B. N., & Marconi, V. C. (2017). Code saturation versus meaning saturation: How many interviews are enough? *Qualitative Health Research*, *27*(4), 591–608. doi:10.1177/1049732316665344

Hoti, E. (2015). The technological, organizational and environmental framework of IS innovation adaption in small and medium enterprises. Evidence from research over the last ten years. *International Journal of Business and Management*, *III*(4), 1–14. doi:10.20472/BM.2015.3.4.001

Hsu, P.-F., Ray, S., & Li-Hsieh, Y.-Y. (2014). Examining cloud computing adoption intention, pricing mechanism, and deployment model. *International Journal of Information Management*, *34*(4), 474–488. doi:10.1016/j.ijinfomgt.2014.04.006

Hughes, D. J., Lee, A., Tian, A. W., Newman, A., & Legood, A. (2018). Leadership, creativity, and innovation: A critical review and practical recommendations. *Leadership Quarterly*, *29*(5), 549–569. doi:10.1016/j.leaqua.2018.03.001

Hyett, N., Kenny, A., & Dickson-Swift, V. (2014). Methodology or method a critical review of qualitative case study reports. *International Journal of Qualitative Studies on Health and Well-Being*, *9*(1), 23606. doi:10.3402/qhw.v9.23606

Iivari, N. (2018). Using member checking in interpretive research practice. *Information Technology & People*, *31*(1), 111–133. doi:10.1108/ITP-07-2016-0168

Ilin, V., Ivetić, J., & Simić, D. (2017). Understanding the determinants of e-business adoption in ERP-enabled firms and non-ERP-enabled firms: A case study of the Western Balkan Peninsula. *Technological Forecasting and Social Change, 125*, 206–223. doi:10.1016/j.techfore.2017.07.025

Jamieson, S. (2016). Analyze qualitative data. *Education for Primary Care*, *27*(5), 1–5. doi:10.1080/14739879.2016.1217430

Jamshidi, P., Pahl, C., Mendonca, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The journey so far and challenges ahead. *IEEE Software*, *35*(3), 24–35. doi:10.1109/MS.2018.2141039

Jenner, B. M., & Myers, K. C. (2019). Intimacy, rapport, and exceptional disclosure: a comparison of in-person and mediated interview contexts. *International Journal of Social Research Methodology*, *22*(2), 165-177. doi:10.1080/13645579.2018.1512694

Jentoft, N., & Olsen, T. S. (2019). Against the flow in data collection: How data triangulation combined with a 'slow' interview technique enriches data. *Qualitative Social Work: Research and Practice*, *18*(2), 179-193. doi:10.1177/1473325017712581

Jespersen, L., & Wallace, C. A. (2017). Triangulation and the importance of establishing valid methods for food safety culture evaluation. *Food Research International*, *100*, 244–253. doi:10.1016/j.foodres.2017.07.009

Jiang, Y., & Adams, B. (2015). Co-evolution of infrastructure and source code - an empirical study. In *2015 IEEE/ACM 12th Working Conference on Mining*

Software Repositories (Vol. 8, pp. 45–55). IEEE. doi:10.1109/MSR.2015.12

Josélyne, M. I., Tuheirwe-Mukasa, D., Kanagwa, B., & Balikuddembe, J. (2018). Partitioning microservices: A domain engineering approach. In *Proceedings of the 2018 International Conference on Software Engineering in Africa - SEiA '18* (Vol. 7, pp. 43–49). New York, NY: ACM Press. doi:10.1145/3195528.3195535

Joslin, R., & Müller, R. (2016). Identifying interesting project phenomena using philosophical and methodological triangulation. *International Journal of Project Management*, *34*(6), 1043–1056. doi:10.1016/j.ijproman.2016.05.005

Kadam, R. A. (2017). Informed consent process: A step further towards making it meaningful! *Perspect Clin Res*, *8*(3), 107–112. doi:10.4103/picr.PICR_147_16

Kallio, H., Pietilä, A. M., Johnson, M., & Kangasniemi, M. (2016). Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing*, *72*(12), 2954–2965. doi:10.1111/jan.13031

Kang, H., Le, M., & Tao, S. (2016). Container and microservice-driven design for cloud infrastructure DevOps. In *2016 IEEE International Conference on Cloud Engineering (IC2E)* (pp. 202–211). IEEE. doi:10.1109/IC2E.2016.26

Killalea, T. (2016). The hidden dividends of microservices. *Communications of the ACM*, *59*(8), 42–45. doi:10.1145/2948985

Knoche, H., & Hasselbring, W. (2018). Using microservices for legacy software modernization. *IEEE Software*, *35*(3), 44–49. doi:10.1109/MS.2018.2141035

Knoche, H., & Hasselbring, W. (2019). Drivers and barriers for microservice adoption-A survey among professionals in Germany. *Enterprise Modelling and Information Systems Architectures (EMISA)*, *14*(1), 1–35. doi:10.18417/emisa.14.1

Kohli, R., & Melville, N. P. (2019). Digital innovation: A review and synthesis. *Information Systems Journal*, *29*(1), 200–223. doi:10.1111/isj.12193

Korstjens, I., & Moser, A. (2018). Series: practical guidance to qualitative research. Part 4: Trustworthiness and publishing. *European Journal of General Practice*, *24*(1), 120–124. doi:10.1080/13814788.2017.1375092

Koul, S., & Eydgahi, A. (2017). A systematic review of technology adoption frameworks and their applications. *Journal of Technology Management & Innovation*, *12*(4), 106–113. doi:10.4067/S0718-27242017000400011

Kratzke, N. (2018). A Brief History of Cloud Application Architectures. *Applied Sciences*, *8*(8), 1368. doi:10.3390/app8081368

Kratzke, N., & Quint, P.-C. C. (2017). Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study. *Journal of Systems and Software*, *126*, 1–16. doi:10.1016/j.jss.2017.01.001

Kruth, J. G. (2014). Five qualitative research approaches and their applications in parapsychology. *Journal of Parapsychology*, *79*(2), 219–233. Retrieved from http://www.rhine.org/what-we-do/journal-of-parapsychology.html

Kurnia, S., Karnali, R. J., & Rahim, M. M. (2015). A qualitative study of business-to-business electronic commerce adoption within the Indonesian grocery industry: A multi-theory perspective. *Information and Management*, *52*(4), 518–536. doi:10.1016/j.im.2015.03.003

Kuusinen, K., Balakumar, V., Jepsen, S. C., Larsen, S. H., Lemqvist, T. A., Muric, A., ... Vestergaard, O. (2018). A Large Agile Organization on Its Journey Towards DevOps. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 60–63). IEEE. doi:10.1109/SEAA.2018.00019

Lai, P. (2017). The literature review of technology adoption models and theories for the novelty technology. *Journal of Information Systems and Technology Management*, *14*(1), 21–38. doi:10.4301/S1807-17752017000100002

Larrucea, X., Santamaria, I., Colomo-Palacios, R., & Ebert, C. (2018). Microservices. *IEEE Software, 35*(3), 96–100. doi:10.1109/MS.2018.2141030

Laukkarinen, T., Kuusinen, K., & Mikkonen, T. (2018). Regulated software meets DevOps. *Information and Software Technology*, *97*, 176–178. doi:10.1016/j.infsof.2018.01.011

Le, D. M., Dang, D. H., & Nguyen, V. H. (2018). On domain driven design using annotation-based domain specific language. *Computer Languages, Systems, and Structures*, *54*, 199–235. doi:10.1016/j.cl.2018.05.001

Leal-Rodríguez, A. L., Eldridge, S., Roldán, J. L., Leal-Millán, A. G., & Ortega-Gutiérrez, J. (2015). Organizational unlearning, innovation outcomes, and performance: The moderating effect of firm size. *Journal of Business Research*, *68*(4), 803–809. doi:10.1016/j.jbusres.2014.11.032

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2020). A Survey of DevOps Concepts and Challenges. *ACM Computing Surveys (CSUR)*, *52*(6), 1–35.

doi:10.1145/3359981

Leppanen, M., Makinen, S., Pagels, M., Eloranta, V., Itkonen, J., Mantyla, M. V., & Mannisto, T. (2015). The highways and country roads to continuous deployment. *IEEE Software*, *32*(2), 64–72. doi:10.1109/MS.2015.50

Leppink, J. (2017). Revisiting the quantitative–qualitative-mixed methods labels: Research questions, developments, and the need for replication. *Journal of Taibah University Medical Sciences*, *12*(2), 97–101. doi:10.1016/j.jtumed.2016.11.008

Leung, D., Lo, A., Fong, L. H. N., & Law, R. (2015). Applying the technology-organization-environment framework to explore ICT initial and continued adoption: An exploratory study of an independent hotel in Hong Kong. *Tourism Recreation Research*, *40*(3), 391–406. doi:10.1080/02508281.2015.1090152

Leung, L. (2015). Validity, reliability, and generalizability in qualitative research. *Journal of Family Medicine and Primary Care*, *4*(3), 324. doi:10.4103/2249-4863.161306

Lewis, J., & Fowler, M. (2014). Microservices. Retrieved April 23, 2019, from https://martinfowler.com/articles/microservices.html

Lewis, J., & Fowler, M. (2016). Microservices guide. Retrieved April 23, 2019, from https://martinfowler.com/microservices/

Lian, J.-W., Yen, D. C., & Wang, Y.-T. (2014). An exploratory study to understand the critical factors affecting the decision to adopt cloud computing in Taiwan hospital. *International Journal of Information Management*, *34*(1), 28–36. doi:10.1016/j.ijinfomgt.2013.09.004

Lilgendahl, J. P., Benet-Martinez, V., Bishop, M., Gilson, K., Festa, L., Levenson, C., & Rosenblum, R. (2018). “So now, I wonder, what am I?”: A narrative approach to bicultural identity integration. *Journal of Cross-Cultural Psychology*, *49*(10), 1596-1624. doi:10.1177/0022022118801555

Lingis, A. (2017). What is given in experience: The phenomenological account. *Qualitative Health Research*, *27*(6), 805–809. doi:10.1177/1049732317698959

Lowe, A., Norris, A. C., Farris, A. J., & Babbage, D. R. (2018). Quantifying thematic saturation in qualitative data analysis. *Field Methods*, *30*(3), 191-207. doi:10.1177/1525822X17749386

Lub, V. (2015). Validity in qualitative evaluation. *International Journal of Qualitative Methods*, *14*(5), 1-8. doi:10.1177/1609406915621406

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*, 1–16. doi:10.1016/j.jss.2019.07.083

Lwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., … Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Information and Software Technology*, *114*, 217–230. doi:10.1016/j.infsof.2019.06.010

MacLennan, E., & Belle, J.-P. (2014). Factors affecting the organizational adoption of service-oriented architecture (SOA). *Information Systems and E-Business Management*, *12*(1), 71–100. doi:10.1007/s10257-012-0212-x

Maher, C., Hadfield, M., Hutchings, M., & de Eyto, A. (2018). Ensuring rigor in qualitative data analysis. *International Journal of Qualitative Methods*, *17*(1), 1-13. doi:10.1177/1609406918786362

Malone, H., Nicholl, H., & Tracey, C. (2014). Awareness and minimization of systematic bias in research. *British Journal of Nursing*, *23*(5), 279–282. doi:10.12968/bjon.2014.23.5.279

Malterud, K., Siersma, V. D., & Guassora, A. D. (2016). Sample size in qualitative interview studies: Guided by information power. *Qualitative Health Research*, *26*(13), 1753–1760. doi:10.1177/1049732315617444

Mamotte, N., & Wassenaar, D. (2015). Measuring voluntariness of consent to research: An instrument review. *Journal of Empirical Research on Human Research Ethics*, *10*(2), 121–131. doi:10.1177/1556264615571552

Martínez-Román, J. A., & Romero, I. (2017). Determinants of innovativeness in SMEs: disentangling core innovation and technology adoption capabilities. *Review of Managerial Science*, *11*(3), 543–569. doi:10.1007/s11846-016-0196-x

Martins, R., Oliveira, T., & Thomas, M. A. (2016). An empirical analysis to assess the determinants of SaaS diffusion in firms. *Computers in Human Behavior*, *62*, 19–33. doi:10.1016/j.chb.2016.03.049

Mason, D. M., & Ide, B. (2014). Adapting qualitative research strategies to technology savvy adolescents. *Nurse Researcher*, *21*(5), 40–45. doi:10.7748/nr.21.5.40.e1241

Matua, G. A. (2015). Choosing phenomenology as a guiding philosophy for nursing research. *Nurse Researcher*, *22*(4), 30–34. doi:10.7748/nr.22.4.30.e1325

McCusker, K., & Gunaydin, S. (2015). Research using qualitative, quantitative or mixed methods and choice based on the research. *Perfusion (United Kingdom)*, *30*(7), 537–542. doi:10.1177/0267659114559116

McDermid, F., Peters, K., Jackson, D., & Daly, J. (2014). Conducting qualitative research in the context of pre-existing peer and collegial relationships. *Nurse Researcher*, *21*(5), 28–33. doi:10.7748/nr.21.5.28.e1232

Molina-Azorin, J. F. (2016). Mixed methods research: An opportunity to improve our studies and our research skills. *European Journal of Management and Business Economics, 25*(2), 37–38. doi:10.1016/j.redeen.2016.05.001

Morris, M. C., & Morris, J. Z. (2016). The importance of virtue ethics in the IRB. *Research Ethics, 12*(4), 201–216. doi:10.1177/1747016116656023

Morse, J. M. (2015). Critical analysis of strategies for determining rigor in qualitative inquiry. *Qualitative Health Research*, *25*(9), 1212–1222. doi:10.1177/1049732315588501

Moser, A., & Korstjens, I. (2017). Series: Practical guidance to qualitative research. Part 1: Introduction. *European Journal of General Practice*, *23*(1), 271–273. doi:10.1080/13814788.2017.1375093

Moser, A., & Korstjens, I. (2018). Series: Practical guidance to qualitative research. Part 3: Sampling, data collection, and analysis. *European Journal of General Practice*, *24*(1), 9–18. doi:10.1080/13814788.2017.1375091

Moskalenko, V. V., & Berezenko, Y. S. (2017). The concept of an architectural solution for the service intended to build an enterprise strategy map. *Bulletin of National Technical University "KhPI." Series: System Analysis, Control and Information*

Technology, *55*, 45–50. doi:10.20998/2079-0023.2017.55.08

Naidu, T., & Prose, N. (2018). Re-envisioning member-checking and communicating results as accountability practice in qualitative research : A south African community-based organization example. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *19*(3), Art. 26. doi:10.17169/fqs-19.3.3153

Newman, S. (2015). *Building microservices: Designing fine-grained systems*. Sebastopol, CA: O'Reilly Media.

Niknejad, N., Razak, A., Hussin, C., Prasetyo, Y. A., Amalia, A., & Fajrillah, N. (2018). Service oriented architecture adoption : A systematic review. *International Journal of Integrated Engineering*, *10*(6), 49–58. Retrieved from http://penerbit.uthm.edu.my/

Nordstrom, S. N. (2015). Not so innocent anymore. *Qualitative Inquiry*, *21*(4), 388–401. doi:10.1177/1077800414563804

Nowell, L. S., Norris, J. M., White, D. E., & Moules, N. J. (2017). Thematic analysis: Striving to meet the trustworthiness criteria. *International Journal of Qualitative Methods*, *16*(1), 1–13. doi:10.1177/1609406917733847

Oates, J. (2015). Use of skype in interviews: The impact of the medium in a study of mental health nurses. *Nurse Researcher*, *22*(4), 13–17. doi:10.7748/nr.22.4.13.e1318

O'Connor, R. V., Elger, P., & Clarke, P. M. (2017a). Continuous software engineering-A microservices architecture perspective. *Journal of Software: Evolution and Process*, *29*(11),1–12. doi:10.1002/smr.1866

Oliveira, T., & Martins, M. F. (2011). Literature review of information technology adoption models at firm level. *Electronic Journal of Information Systems Evaluation*, *14*(1), 110–121. Retrieved from http://www.ejise.com

Oliveira, T., Thomas, M., & Espadanal, M. (2014). Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. *Information and Management*, *51*(5), 497–510. doi:10.1016/j.im.2014.03.006

Ortlipp, M. (2008). Keeping and using reflective journals in the qualitative research process. *The Qualitative Report*, *13*(4), 695–705. Retrieved from http://nsuworks.nova.edu/tqr/

Ose, S. O. (2016). Using excel and word to structure qualitative data. *Journal of Applied Social Science*, *10*(2), 147–162. doi:10.1177/1936724416664948

Øye, C., Sørensen, N. Ø., & Glasdam, S. (2016a). Qualitative research ethics on the spot: Not only on the desktop. *Nursing Ethics*, *23*(4), 455–464. doi:10.1177/0969733014567023

Pahl, C., Brogi, A., Soldani, J., & Jamshidi, P. (2019). Cloud container technologies: a state-of-the-art review. *IEEE Transactions on Cloud Computing, 7*(3), 677–692. doi:10.1109/TCC.2017.2702586

Pahl, C., & Jamshidi, P. (2016). Microservices: A systematic mapping study. In *Proceedings of the 6th International Conference on Cloud Computing and Services Science* (pp. 137–146). Scitepress - Science and Technology Publications. doi:10.5220/0005785501370146

Palinkas, A., Horwitz, S., Green, C., Wisdom, J., Duan, N., & Hoagwood, K. (2015).

HHS Public Access. *Administration and Policy in Mental Health and Mental Health*, *42*(5), 533–544. doi:10.1161/CIRCRESAHA.116.303790

Pallis, G., Trihinas, D., Tryfonos, A., & Dikaiakos, M. (2018). DevOps as a service: Pushing the boundaries of microservice adoption. *IEEE Internet Computing*, *22*(3), 65–71. doi:10.1109/MIC.2018.032501519

Pautasso, C., Zimmermann, O., Amundsen, M., Lewis, J., & Josuttis, N. (2017). Microservices in practice, part 1: Reality check and service design. *IEEE Software*, *34*(1), 91–98. doi:10.1109/MS.2017.24

Percy, W. H., Kostere, K., & Kostere, S. (2015). Generic qualitative research in psychology. *The Qualitative Report Qualitative Research, Thematic Analysis Generic Qualitative Research Basic Qualitative Research*, *20*(5), 76–85. Retrieved from http://www.nova.edu/ssss

Petrasch, R. (2017). Model-based engineering for microservice architectures using Enterprise Integration Patterns for inter-service communication. In *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1–4). IEEE. doi:10.1109/JCSSE.2017.8025912

Petrova, E., Dewing, J., & Camilleri, M. (2016). Confidentiality in participatory research: Challenges from one study. *Nursing Ethics*, *23*(4), 442–454. doi:10.1177/0969733014564909

Pietkiewicz, I., & Smith, J. A. (2014). A practical guide to using Interpretative Phenomenological Analysis in qualitative research psychology. *Czasopismo Psychologiczne Psychological Journal*, *20*(1), 7–14. doi:10.14691/CPPJ.20.1.7

Pozdniakova, O., & Zeika, D. M. (2017). Systematic Literature Review of the Cloud-ready Software Architecture. *Baltic J. Modern Computing*, 5(1), 124–135. doi:10.22364/bjmc.2017.5.1.08

Premkumar, G., & Ramamurthy, K. (1995). The role of interorganizational and organizational factors on the decision mode for adoption of interorganizational systems. *Decision Sciences*, *26*(3), 303–336. doi:10.1111/j.1540-5915.1995.tb01431.x

Quick, J., & Hall, S. (2015a). Part one: An introduction to the research process. *Journal of Perioperative Practice*, *25*(4), 78–82. doi:10.1177/175045891502500404

Quick, J., & Hall, S. (2015b). Part two: Qualitative research. *Journal of Perioperative Practice*, *25*(7–8), 129–133. doi:10.1177/1750458915025007-803

Rademacher, F., Sorgalla, J., & Sachweh, S. (2018). Challenges of domain-driven microservice design: A model-driven perspective. *IEEE Software*, *35*(3), 36–43. doi:10.1109/MS.2018.2141028

Rahayu, R., & Day, J. (2015). Determinant factors of e-commerce adoption by SMEs in developing country: Evidence from Indonesia. *Procedia - Social and Behavioral Sciences*, *195*, 142–150. doi:10.1016/j.sbspro.2015.06.423

Ramanathan, L., & Krishnan, S. (2015). An empirical investigation into the adoption of open source software in information technology outsourcing organizations. *Journal of Systems and Information Technology*, *17*(2), 167–192. doi:10.1108/JSIT-10-2014-0070

Ranney, M. L., Meisel, Z. F., Choo, E. K., Garro, A. C., Sasson, C., & Morrow Guthrie,

K. (2015). Interview-based qualitative research in emergency care part II: Data collection, analysis, and results reporting. *Academic Emergency Medicine*, *22*(9), 1103–1112. doi:10.1111/acem.12735

Renz, S. M., Carrington, J. M., & Badger, T. A. (2018). Two strategies for qualitative content analysis: An intramethod approach to triangulation. *Qualitative Health Research*, *28*(5), 824–831. doi:10.1177/1049732317753586

Reynolds, J. D., Choi, T., & Lee, M. (2018). Ethical and methodological issues resulting from recording lapses in qualitative research. *The Qualitative Report 2018*, *23*(7), 1509–1514. Retrieved from https://nsuworks.nova.edu/tqr

Riese, J. (2019). What is 'access' in the context of qualitative research? *Qualitative Research*, *19*(6), 669-684. doi:10.1177/1468794118787713

Robertson, J. H., & Thomson, A. M. (2014). A phenomenological study of the effects of clinical negligence litigation on midwives in England: The personal perspective. *Midwifery*, *30*(3), 121–130. doi:10.1016/j.midw.2013.12.003

Robins, C. S., & Eisen, K. (2017). Strategies for the effective use of NVivo in a large-scale study: Qualitative analysis and the repeal of don't ask, don't tell. *Qualitative Inquiry*, *23*(10), 768–778. doi:10.1177/1077800417731089

Robinson, O. C. (2014). Sampling in interview-based qualitative research: A theoretical and practical guide. *Qualitative Research in Psychology*, *11*(1), 25–41. doi:10.1080/14780887.2013.801543

Rogers, E. M. (1983). *Diffusion of innovations* (3rd ed.). New York, NY: The Free Press.

Rogers, E. M. (2003). *Diffusion of innovations* (5th ed.). New York, NY: The Free Press.

Rosiek, J. L., & Snyder, J. (2018). Narrative inquiry and new materialism: Stories as (not necessarily benign) Agents. *Qualitative Inquiry*. doi:10.1177/1077800418784326

Russo, D., & Ciancarini, P. (2017). Towards antifragile software architectures. *Procedia Computer Science*, *109*, 929–934. doi:10.1016/j.procs.2017.05.426

Saunders, B., Kitzinger, J., & Kitzinger, C. (2015). Anonymizing interview data: challenges and compromise in practice. *Qualitative Research*, *15*(5), 616–632. doi:10.1177/1468794114550439

Schermann, G., Cito, J., Leitner, P., Zdun, U., & Gall, H. C. (2018). We're doing it live: A multi-method empirical study on continuous experimentation. *Information and Software Technology*, *99*, 41-57. doi:10.1016/j.infsof.2018.02.010

Schwartz, A. (2017). Microservices. *Informatik-Spektrum*, *40*(6), 590–594. doi:10.1007/s00287-017-1078-6

Seitz, S. (2016). Pixilated partnerships, overcoming obstacles in qualitative interviews via Skype: a research note. *Qualitative Research*, *16*(2), 229–235. doi:10.1177/1468794115577011

Senyo, P. K., Addae, E., & Boateng, R. (2018). Cloud computing research: A review of research themes, frameworks, methods and future research directions. *International Journal of Information Management*, *38*(1), 128–139. doi:10.1016/j.ijinfomgt.2017.07.007

Senyo, P. K., Effah, J., & Addae, E. (2016). Preliminary insight into cloud computing adoption in a developing country. *Journal of Enterprise Information Management*, *29*(4), 505–524. doi:10.1108/JEIM-09-2014-0094

Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous integration, delivery, and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access*, *5*, 3909–3943. doi:10.1109/ACCESS.2017.2685629

Sharma, S., & Rai, A. (2015). Adopting IS process innovations in organizations: The role of IS leaders' individual factors and technology perceptions in decision making. *European Journal of Information Systems*, *24*(1), 23–37. doi:10.1057/ejis.2013.24

Skea, D. (2016). Phenomenological inquiry and psychological research in caring and quality of life contexts : Acknowledging the invisible principles. *International Journal of Caring Sciences*, *9*(3), 1134–1146. Retrieved from http://www.internationaljournalofcaringsciences.org

Smite, D., Moe, N. B., Levinta, G., & Floryan, M. (2019). Spotify guilds: How to succeed with knowledge sharing in large-scale agile organizations. *IEEE Software*, *36*(2), 51–56. doi:10.1109/MS.2018.2886178

Smith, P. R. (2018). Collecting sufficient evidence when conducting a case study. *The Qualitative Report*, *23*(5), 1048–1054. Retrieved from https://nsuworks.nova.edu/tqr/vol23/iss5/2

Soldani, J., Tamburri, D. A., & Van Den Heuvel, W.-J. (2018). The pains and gains of microservices: A Systematic grey literature review. *Journal of Systems and Software*, *146*, 215–232. doi:10.1016/j.jss.2018.09.082

Sorgalla, J., Rademacher, F., Sachweh, S., & Zündorf, A. (2018). On collaborative model-driven development of microservices, 1–8. Retrieved from http://arxiv.org/abs/1805.01176

Sotiriadou, P., Brouwers, J., & Le, T. A. (2014). Choosing a qualitative data analysis tool: A comparison of NVivo and Leximancer. *Annals of Leisure Research*, *17*(2), 218–234. doi:10.1080/11745398.2014.902292

Soto-Acosta, P., Popa, S., & Martinez-Conesa, I. (2018). Information technology, knowledge management and environmental dynamism as drivers of innovation ambidexterity: a study in SMEs. *Journal of Knowledge Management*, *22*(4), 824–849. doi:10.1108/JKM-10-2017-0448

Sperber, S., & Linder, C. (2018). The impact of top management teams on firm innovativeness: a configurational analysis of demographic characteristics, leadership style, and team power distribution. *Review of Managerial Science*, *12*(1), 285–316. doi:10.1007/s11846-016-0222-z

Srivastava, J., & Nanath, K. (2017). Adoption of cloud computing in UAE. *International Journal of Information Systems in the Service Sector*, *9*(4), 1–20. doi:10.4018/IJISSS.2017100101

Ståhl, D., Mårtensson, T., & Bosch, J. (2017). The continuity of continuous integration: Correlations and consequences. *Journal of Systems and Software*, *127*, 150–167. doi:10.1016/j.jss.2017.02.003

Steininger, K., & Stiglbauer, B. (2015). EHR acceptance among Austrian resident doctors. *Health Policy and Technology*, *4*(2), 121–130. doi:10.1016/j.hlpt.2015.02.003

Stern, A. D. (2017). Innovation under regulatory uncertainty: Evidence from medical technology. *Journal of Public Economics*, *145*, 181–200.

doi:10.1016/j.jpubeco.2016.11.010

Sun, S., Cegielski, C. G., Jia, L., & Hall, D. J. (2018). Understanding the factors affecting the organizational adoption of big data. *Journal of Computer Information Systems*, *58*(3), 193–203. doi:10.1080/08874417.2016.1222891

Suram, S., MacCarty, N. A., & Bryden, K. M. (2018). Engineering design analysis utilizing a cloud platform. *Advances in Engineering Software, 115*, 374–385. doi:10.1016/j.advengsoft.2017.10.004

Suzuki, O. (2018). Enabling or constraining? Unraveling the influence of organizational slack on innovation. *Industrial and Corporate Change*, *27*(3), 555–575. doi:10.1093/icc/dtx046

Taguchi, N. (2018). Description and explanation of pragmatic development: Quantitative, qualitative, and mixed methods research. *System*, *75*, 23–32. doi:10.1016/j.system.2018.03.010

Taherizadeh, S., & Grobelnik, M. (2020). Key influencing factors of the Kubernetes auto-scaler for computing-intensive microservice-native cloud-based applications. *Advances in Engineering Software, 140*. doi:10.1016/j.advengsoft.2019.102734

Taibi, D., & Lenarduzzi, V. (2018). On the definition of microservice bad smells. *IEEE Software*, *35*(3), 56–62. doi:10.1109/MS.2018.2141031

Tarhini, A., Arachchilage, N. A. G., Masa'deh, R., & Abbasi, M. S. (2015). A critical review of theories and models of technology adoption and acceptance in information system research. *International Journal of Technology Diffusion*, *6*(4), 58–77. doi:10.4018/IJTD.2015100104

Tarmizi, A., & Shanudin, M. (2017). A method for analyzing and designing microservice holistically. *International Journal of Advanced Computer Science and Applications*, *8*(12). doi:10.14569/IJACSA.2017.081236

Teixeira Vinci, A. L., Lopes Rijo, R. P. C., de Azevedo Marques, J. M., & Alves, D. (2017). Development and proposal of a reference tool for semi-structured interviews for the characterization of the management in mental health networks. *Procedia Computer Science*, *121*, 511–518. doi:10.1016/j.procs.2017.11.068

Terzić, B., Dimitrieski, V., Kordić, S., Milosavljević, G., & Luković, I. (2018). Development and evaluation of MicroBuilder: a Model-Driven tool for the specification of REST Microservice Software Architectures. *Enterprise Information Systems, 12*(8-9), 1034-1057. doi:10.1080/17517575.2018.1460766

Thompson Long, B., & Hall, T. (2018). Educational narrative inquiry through design-based research: Designing digital storytelling to make alternative knowledge visible and actionable. *Irish Educational Studies, 37*(2), 205–225. doi:10.1080/03323315.2018.1465836

Thönes, J. (2015). Microservices. *IEEE Software*, *32*(1), 116–116. doi:10.1109/MS.2015.11

Thong, J. Y. (1999). An integrated model of information systems adoption in small businesses. *Journal of Management Information Systems*, *15*(4), 187–214. doi:10.1080/07421222.1999.11518227

Tornatzky, L. G., & Fleischer, M. (1990). *The process of technological innovation*. Lexington, MA: Lexington Books.

Tornatzky, L. G., & Klein, K. J. (1982). Innovation characteristics and innovation adoption-implementation: A meta-analysis of findings. *IEEE Transactions on Engineering Management*, *EM-29*(1), 28–45. doi:10.1109/TEM.1982.6447463

Trnka, S. (2017). The fifty-minute ethnography: Teaching theory through fieldwork. *The Journal of Effective Teaching*, *17*(1), 28–34. Retrieved from https://eric.ed.gov

Tuck, E., & McKenzie, M. (2015). Relational validity and the "where" of inquiry. *Qualitative Inquiry*, *21*(7), 633–638. doi:10.1177/1077800414563809

Uludag, Ö., Hauder, M., Kleehaus, M., Schimpfle, C., & Matthes, F. (2018). Supporting Large-Scale Agile Development with Domain-Driven Design. In *Lecture Notes in Computer Science* (Vol. 1, pp. 232–247). doi:10.1007/978-3-319-91602-6_16

United States. (1979). *The Belmont report. Ethical principles and guidelines for the protection of human subjects of research.* Bethesda, Md. Retrieved from https://www.hhs.gov/ohrp/regulations-and-policy/belmont-report/index.html

Vanacker, T., Collewaert, V., & Zahra, S. A. (2017). Slack resources, firm performance, and the institutional context: Evidence from privately held European firms. *Strategic Management Journal*, *38*(6), 1305–1326. doi:10.1002/smj.2583

Varpio, L., Ajjawi, R., Monrouxe, L. V., O'Brien, B. C., & Rees, C. E. (2017). Shedding the cobra effect: Problematising thematic emergence, triangulation, saturation, and member checking. *Medical Education*, *51*(1), 40–50. doi:10.1111/medu.13124

Vasquez-Tokos, J. (2017). "If I can offer you some advice": Rapport and data collection in interviews between adults of different ages. *Symbolic Interaction*, *40*(4), 463–

482. doi:10.1002/symb.305

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, *27*(3), 425–478. doi:10.2307/30036540

Vicary, S., Young, A., & Hicks, S. (2017). A reflective journal as learning process and contribution to quality and validity in interpretative phenomenological analysis. *Qualitative Social Work: Research and Practice*, *16*(4), 550–565. doi:10.1177/1473325016635244

Vohra, V. (2014). Using the multiple case study design to Decipher contextual leadership behaviors in Indian organizations. *Electronic Journal of Business Research Methods*, *12*(1), 54–65. Retrieved from http://www.ejbrm.com

Waardenburg, G., & Vliet, H. (2013). When agile meets the enterprise. *Information and Software Technology*, *55*(12), 2154–2171. doi:10.1016/j.infsof.2013.07.012

Wan, X., Guan, X., Wang, T., Bai, G., & Choi, B. Y. (2018). Application deployment using Microservice and Docker containers: Framework and optimization. *Journal of Network and Computer Applications*, *119*, 97–109. doi:10.1016/j.jnca.2018.07.003

Wang, Y. S., Li, H. T., Li, C. R., & Zhang, D. Z. (2016). Factors affecting hotels' adoption of mobile reservation systems: A technology-organization-environment framework. *Tourism Management*, *53*, 163–172. doi:10.1016/j.tourman.2015.09.021

Watkins, D. C. (2017). Rapid and rigorous qualitative data analysis: The "RADaR" technique for applied research. *International Journal of Qualitative Methods*, *16*(1), 1–9. doi:10.1177/1609406917712131

Williams, C. L., Sica, J. C., Killen, R. T., & Balis, U. G. J. (2016). The growing need for microservices in bioinformatics. *Journal of Pathology Informatics, 7*(1), 45-48. doi:10.4103/2153-3539.194835

Wilson, A. (2014). Being a practitioner: an application of Heidegger's phenomenology. *Nurse Researcher*, *21*(6), 28–33. doi:10.7748/nr.21.6.28.e1251

Yamagata-Lynch, L. C., Do, J., Deshpande, D., Skutnik, A. L., Murphy, B. K., & Garty, E. (2017). Narrative inquiry with activity systems: A story about net neutrality. *International Journal of Qualitative Methods*, *16*(1), 1–11. doi:10.1177/1609406917704352

Yang, Z., Sun, J., Zhang, Y., & Wang, Y. (2015). Understanding SaaS adoption from the perspective of organizational users: A tripod readiness model. *Computers in Human Behavior, 45,* 254–264. doi:10.1016/j.chb.2014.12.022

Yazan, B. (2015). Three approaches to case study methods in education : Yin, Merriam, and Stake. *The Qualitative Report*, *20*(2), 134–152. Retrieved from http://nsuworks.nova.edu/tqr/vol20/iss2/12

Yin, R. K. (2014). *Case study research: Design and methods* (4th ed.). Thousand Oaks, CA: SAGE Publications

Zamawe, F. (2015). The implication of using NVivo software in qualitative data analysis: Evidence-based reflections. *Malawi Medical Journal*, *27*(1), 13. doi:10.4314/mmj.v27i1.4

Zewen, C., Xin, L., & Hongjun, C. (2017). Research on the influence of institutional pressures on green innovation strategy. *IOP Conference Series: Earth and Environmental Science*, *94*. doi:10.1088/1755-1315/94/1/012162

Zhang, H., Li, S., Jia, Z., Zhong, C., & Zhang, C. (2019). Microservice Architecture in Reality: An Industrial Inquiry. In *2019 IEEE International Conference on Software Architecture (ICSA)* (pp. 51–60). IEEE. doi:10.1109/ICSA.2019.00014

Zhao, J. T., Jing, S. Y., & Jiang, L. Z. (2018). Management of API Gateway Based on Micro-service Architecture. *Journal of Physics: Conference Series, 1087*. doi:10.1088/1742-6596/1087/3/032032

Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and its practices. *IEEE Software*, *33*(3), 32–34. doi:10.1109/MS.2016.81

Zimmermann, O. (2016). Microservices tenets. *Computer Science - Research and Development*, *32*(3–4), 301–310. doi:10.1007/s00450-016-0337-0

Zimmermann, O., Miksovic, C., & Küster, J. M. (2012). Reference architecture, metamodel, and modeling principles for architectural knowledge management in information technology services. *Journal of Systems and Software*, *85*(9), 2014–2033. doi:10.1016/j.jss.2012.05.003

Zurlo, F., & Cautela, C. (2014). Design strategies in different narrative frames. *Design Issues*, *30*(1), 19–35. doi:10.1162/DESI_a_00246

Appendix A: Projecting Human Research Participants Certificate of Completion

Certificate of Completion

The National Institutes of Health (NIH) Office of Extramural Research certifies that **Walter Zrzavy** successfully completed the NIH Web-based training course "Protecting Human Research Participants".

Date of completion: 03/17/2017.

Certification Number: 2349535.

Appendix B: Interview Questions

Initial Questions

1. What is your current position and role?
2. How long have you been in this or similar position?
3. How long do you work in this organization?
4. How long is your experience in working with systems based on the microservice architecture concept?
5. How long is your experience in domain-driven design?
6. How long is your experience in DevOps?
7. What is your understanding of the microservice architecture concept? Please explain.

Interview Questions

1. What inefficiencies or technical debt do you and your team experience using the domain-driven design approach in the development of MSA-based applications?
2. What domain-driven-design strategies do you use to identify, reduce, or prevent inefficiencies or technical debt of MSA-based applications?
3. Which domain-driven-design strategies are most effective in reducing the inefficiencies of MSA-based applications?
4. What challenges do you and your team experience using DevOps for MSA-based applications?

5. What DevOps practices do you and your team apply for MSA-based applications?
6. Which DevOps strategies were most effective in reducing the inefficiencies of MSA-based applications?
7. What, if any, other inefficiencies or challenges do you experience in the adoption of MSA systems?
8. What strategies do you apply for adopting MSA-based system in your organization?
9. What factors do you apply in selecting the domain-driven design strategies and DevOps practices for the MSA system adoption?

Possible follow up questions

1. What inefficiencies or technical debt had the highest impact on your organization?
2. What domain-driven-design strategy had the highest impact on the efficiency of your MSA system?
3. What DevOps challenges with MSA systems had the highest impact on your organization?
4. What DevOps practice had the highest impact on the efficiency of your MSA system?
5. How long did you need to implement the strategies?
6. What strategy had the most positive effect on your MSA adoption?
7. What is the ranking of these factors by you and your organization?

Wrap up questions

1. If you had to describe your strategies in four to six words, what are your key concepts or elements of the strategy?
2. What additional information would you like to share about this topic?

Appendix C: Interview Protocol

Organization	
Participant ID	
Location	
Date/Time	
Introduction to the interview	Thank you for your time and for participating in this interview. My name is Walter Zrzavy, and I am in the program of Doctor of Information Technology at Walden University. My background is in information technology consulting for infrastructure services and software development. I am in the industry since 1992 in various technical roles and positions.
Explain the purpose of the interview	The purpose of this study is to explore the domain-driven design strategies and DevOps practices used by practitioners and organizations to reduce the inefficiencies in the adoption of MSA systems. While organizations realize benefits through the adoption of MSA, practitioners report of challenges in the adoption caused by domain-driven design and DevOps. Therefore, this study aims to explore the strategies used by practitioners in addressing these challenges.
Explain the reason for participating	Your responses to the questions and sources that you may share will support my study in partial fulfillment of the degree of Doctor of Information Technology from Walden University.
Describe the risks and benefits of participation	This interview would not pose any risks beyond those of a typical daily life. There are no direct benefits to you. The information may add to the research and professional knowledge on MSA adoption using domain-driven design strategies and DevOps practices to reduce inefficiencies.
Discuss the right to privacy	I am adhering to Walden University's ethical research standards and your right to privacy. You can withdraw from this interview and research without any consequences. You are free to refuse to answer any question if you are not comfortable providing the information. Are you ok to continue? I am requesting your permission to start the audio recording of this interview and document this entire interview using notes. I will use your participant ID {X} and ask you to reconfirm your permission to record and documentation of the interview. Do you agree to start the recording of the interview now?
Start recording of the interview	My name is Researcher Zrzavy, and I am in a Skype session with the participant {Participant ID}. Today's date is dd/mm/yy, and the time is hh:mm. Would you please confirm that I have provided the purpose of this research, the reason for your participation, your

	benefits of participation, and that you approve the electronic recording and taking notes during this interview?
Address the confidentiality of this interview	This interview is completely voluntary, and you may withdraw from the interview or research participation at any time without stating a reason and consequences. You are free to refuse to answer any question if you are not comfortable providing the information. All information that you share will be kept strictly confidential and will not be disclosed to your employer or others. I request that you use synonyms for individuals and organizations and do not use names of individuals, organizations, or aspects that would allow others to identify the individual or organization. In the case that names or details are mentioned during the interview, I will remove those from the transcript and study report. Furthermore, I request that you do not share or discuss your participation in this research until the study is finished. Any information shared will be used for this study only. The information will be merged with data from other participants for analysis as an anonymized report in a doctoral study. The doctoral study may be published electronically. I will password protect, encrypt all research data, store the records in the safe for five years, which only I have access to the data. I will safely destroy the records after five years, starting from the publication date of the doctoral study.
Checkpoint before the interview	Do you have any questions for me, or do you want to withdraw your participation? If no, would you be ok to start the interview?
The start of the interview	The interview is semistructured with a set of questions about your experience as a practitioner on domain-driven design, DevOps, and MSA adoption. I appreciate it if you could answer my questions with honest thoughts and share as much as detailed information as possible. I may ask you follow-up questions on parts of your responses and would appreciate it if you could provide your point of view and thoughts.
Initial probing questions and warm-up	1. What is your current position and role? 2. How long have you been in this or similar position? 3. How long do you work in this organization? 4. How long is your experience in working with systems based on the microservice architecture concept? 5. How long is your experience in domain-driven design? 6. How long is your experience in DevOps? 7. What is your understanding of the microservice architecture concept? Please explain.

Targeted concept questions	1. What inefficiencies or technical debt do you and your team experience using the domain-driven design approach in the development of MSA-based applications? 2. What domain-driven design strategies do you use to identify, reduce, or prevent inefficiencies or technical debt of MSA-based applications? 3. Which domain-driven design strategies were most effective in reducing the inefficiencies of MSA-based applications? 4. What challenges do you and your team experience using DevOps for MSA-based applications? 5. What DevOps practices do you and your team apply for MSA-based applications? 6. Which DevOps strategies were most effective in reducing the inefficiencies of MSA-based applications? 7. What, if any, other inefficiencies or challenges do you experience in the adoption of MSA systems? 8. What strategies do you apply for adopting MSA-based system in your organization? 9. What factors do you apply in selecting the domain-driven design strategies and DevOps practices for the MSA system adoption?
Targeted follow-up questions	1. What inefficiencies or technical debt had the highest impact on your organization? 2. What domain-driven design strategy had the highest impact on the efficiency of your MSA system? 3. What DevOps challenges with MSA systems had the highest impact on your organization? 4. What DevOps practice had the highest impact on the efficiency of your MSA system? 5. How long did you need to implement the strategies? 6. What strategy had the most positive effect on your MSA adoption? 7. What is the ranking of these factors by you and your organization?
Targeted wrap-up questions	1. If you had to describe your strategies in four to six words, what are your key concepts or elements of the strategies? 2. What additional information would you like to share about this topic?
Follow-up for member checking	Thank you so much for your time today. I would like to schedule a follow-up interview to ensure I interpreted your information correctly. I would send you a copy of the interpretation via email in a couple of days before the follow-up session to optimize your availability. Would this be ok for you?
Conclusion	Thank you very much again for your time to participate in this study.