iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

DevOps and Information Technology Service Management: A Problem Management Case Study

Daniel Maass Adriano

Master’s in Information Systems Management

Supervisor:
PhD, Rúben Filipe de Sousa Pereira, Assistant Professor
ISCTE

Co-Supervisor:
PhD, Rafael Almeida
Neobrain

October, 2021

Department of Information Science and Technology

DevOps and Information Technology Service Management: A Problem Management Case Study

Daniel Maass Adriano

Master's in Information Systems Management

Supervisor:
PhD, Rúben Filipe de Sousa Pereira, Assistant Professor
ISCTE

Co-Supervisor:
PhD, Rafael Almeida
Neobrain

October, 2021

Acknowledgements

The writing of this thesis would hardly have been possible without the generous contribution and support of many individuals. I am profoundly grateful to each of them.

First, I must thank my supervisor, Prof. Dr. Rúben Filipe de Sousa Pereira. His constant availability, encouragement and insight brought over the course of this project cannot be overstated. For bringing a clearer vision to this work, and investing so much time into guiding me, thank you.

I would also thank my co-supervisor, Dr. Rafael Almeida, as well as the wonderful professors I had the pleasure of learning from over the past years. Of these, I would like to mention Prof. Dr. Bráulio Alexandre Barreira Alturas and Prof. Dr. Abílio Oliveira, whose guidance was very valuable in the early stages of this research.

I have to thank my incredible co-workers for their time, their kind and heartening words, and for sharing with me their own experience. The support of these esteemed friends was crucial to me.

Lastly, I must thank my family and dear parents. My dad, for his unconditional support, motivating me in the pursuit of my goals; and my wonderful mom, constantly reminding me of the importance of patience, of *one day at a time*.

I am so grateful to all of you.

Resumo

A utilização de metodologias DevOps é hoje uma característica predominante de organizações envolvidas no desenvolvimento e manutenção de sistemas de Tecnologia e Informação. Apesar da crescente produção de literatura a examinar práticas DevOps, existe muito território por explorar referente às suas ramificações a nível operacional. Isto é particularmente notável quando se consideram potenciais interações com frameworks de ITSM como o ITIL, que governam Operações. Esta pesquisa tem como objetivo estabelecer quais princípios e práticas DevOps podem ser aplicadas na Gestão de Problemas, um processo central para a Gestão de Serviços. Especificamente, exploramos quais práticas DevOps podem ser utilizadas ao longo do ciclo de vida de um Problema, tal como que benefícios poderão resultar da sua aplicação. Um caso de estudo exploratório foi realizado com a participação de Gestores de Problemas a operar num ambiente DevOps. Três métodos de recolha de dados foram aplicados: Entrevistas semiestruturadas, onde participantes descreveram a sua experiência e conhecimento em relação a DevOps e Gestão de Problemas; análise documental e observação, onde processos operacionais foram examinados; e uma discussão em grupo onde resultados do estudo foram discutidos e sistematizados. Esta investigação indica que práticas DevOps tem variados níveis de significância para um processo de Gestão de Problemas. Práticas associadas ao planeamento contínuo e colaboração tendem a ter maior significância no ciclo de vida de um Problema, com potencial para gerar benefícios como a mais rápida identificação de Problemas, maior qualidade na análise de causa, e melhorias nos tempos de resolução. As conclusões apresentadas neste estudo trazem benefícios tanto para académicos, expandindo o corpo de conhecimento disponível sobre o tema, como para profissionais, considerando a sua natureza prática e aplicável. Direções para trabalho futuro são também apresentadas.

Palavras-Chave: DevOps, ITSM, ITIL, Gestão de Problemas.

Abstract

The use of DevOps is a predominant attribute of businesses engaged in the development and maintenance of Information Technology systems. Although literature exploring DevOps practices has expanded, there is still much unexplored territory on its operational ramifications. This is particularly observed when considering their potential impact on ITSM frameworks such as ITIL, which governs Operations. This research aims to establish how DevOps principles and practices can be applied to Problem Management, a core Service Management process. Specifically, it explores which DevOps practices may be used throughout the Problem lifecycle, as well as benefits which may result from them. An exploratory case study was carried out with the participation of Problem Managers operating in a DevOps environment. Three data collection methods were applied: Semi structured interviews, in which participants described their experience and insight in relation to DevOps and Problem Management; documental analysis and observation, where processes and workflows were examined; and a focus group exercise in which study outcomes were discussed and systematized. This research indicates that DevOps practices have varying degrees of significance for a Problem Management process. Practices associated with continuous planning and collaboration are prone to having greater significance in a Problem lifecycle, with the potential of enabling benefits such as quicker Problem identification, higher quality Root Cause Analysis, and improved resolution times. The novelty of insight gathered in this study benefits both academics, through its contribution to an expanding body of knowledge, and professionals, considering the practical and applicable nature of findings. Future work is also presented.

Keywords: DevOps, ITSM, ITIL, Problem Management.

Index

Table Index

Figure Index

Abbreviations and Acronyms

CAMS	–	Culture, Automation, Measurement, Sharing
CD	–	Continuous Deployment
CDE	–	Continuous Delivery
CI	–	Continuous Integration
CMMI	–	Capability Maturity Model Integration
COBIT	–	Control Objectives for Information and Related Technologies
CS	–	Case study
DevOps		Development and Operations
DSR	–	Design Science Research
IM	–	Incident Management
ISO	–	International Organization for Standardization
IT	–	Information Technology
ITIL	–	Information Technology Infrastructure Library
ITSM	–	Information Technology Service Management
PM	–	Problem Management
RCA	–	Root Cause Analysis
RQ	–	Research Question
SLR	–	Systematic Literature Review
UK	–	United Kingdom
USA	–	United States of America

Chapter 1 – Introduction

The increased introduction of Information Technology (IT) resources into modern-day products and services has profoundly transformed the nature of business worldwide (Limanto et al., 2017). One could say that this development triggered the start of a new "information age" (Varga et al., 2019) in which the proper management of knowledge, processes and service delivery are present at the very center of organizational priorities, being key for success (Alsolamy et al., 2014; Park et al., 2006). Technological developments have driven up the investment on IT, as organizations strive to capture the benefits resulting from innovation as well as protect themselves against new and emerging threats (Benitez et al., 2018; Luo et al., 2016). In an effort to cope with the constant competition present in increasingly dynamic markets (Badinelli et al., 2012) organizations have developed and put in place equally "complex and dynamic IT systems to support their business processes" (Jamous et al., 2016; Soni, 2015).

Due to changing market demands, the very methodologies by which software is delivered have evolved (Virmani, 2015). As stated by Šćekić et al., (2018), ever-changing business needs, coupled with the present expediency requirements for introducing software into the market, have "created a paradigm shift towards a 3rd generation Software Development philosophy called DevOps", an acronym for software Development and IT Operations (See also Laukkanen et al., 2017).

Many have defined the new sets of practices contemplated in DevOps as the merging of people, process, and product, aimed at delivering greater value to customers (de Kort, 2016) through quicker, yet equally competent and reliable, development and release cycles (Farroha & Farroha, 2014; Mohan et al., 2018). As such, it should not come as a surprise why these have become so increasingly popular in present-day software development (Ståhl et al., 2017).

The emerging DevOps philosophy not only relates to software development standards but plays a significant role in the level and frequency of interaction between development and operations staff (Aiello & Sachs, 2016; Guerriero et al., 2015). Literature examining the processes by which DevOps is adopted frequently mention collaborative links being formed between said teams; these reportedly result into greater knowledge sharing (Senapathi et al., 2018), improved risk assessment and error detection (Gupta et al., 2019) as well as the reduction of deployment cycle duration (Kuusinen et al., 2018). This has led to new research being performed on the potential challenges, risks, and benefits of the interaction between DevOps practices and Information Technology Service Management (ITSM) frameworks, which influence operations (Pilorget & Schell, 2018). It is through the “specialized organizational capabilities” that ITSM offers, in the form of processes, functions and roles, that value can be consistently generated and delivered (Cartlidge & Lillycrop, 2009).

The outcomes of DevOps practices in organizations operating via ITSM have been largely unexplored (Abdelkebir et al., 2017; Kamuto & Langerman, 2017). Amongst established frameworks such as COBIT and ISO/IEC (Sukmandhani et al., 2017), the Information Technology Infrastructure Library (ITIL) stands out as the most widely adopted ITSM system (Sharifi et al., 2008). Prevalently used ITIL processes such as Incident and Problem Management (PM) are of great significance for businesses in virtually any sector (Krishna Kaiser, 2018). As such, one can perceive an opportunity of delving deeper into the potential links between these key processes and the DevOps methodology.

Research has been performed to understand how DevOps practices may have a role in executing the Incident Management process (Faustino, 2018). We will give continuity to this effort, focusing our analysis on exploring and understanding possible interactions with the PM process. To achieve this objective, the investigation will rely on the design science research (DSR) methodology. It will also be complemented with a systematic literature review (SLR) and a case study.

The study is organized into 6 chapters, this first of which provides background and contextualization to the matters being investigated. It introduces motivations, the research question, main objectives, and a description of the dissertation structure. In the second chapter we provide a brief literature review on DevOps, ITSM frameworks, as well as the PM process; we also review currently existing literature on DevOps and ITSM integration. The third and fourth chapter are dedicated to presenting the research methodology we relied on. The practices followed for the capturing and treatment of case study data, and information regarding participants, are included therein. Analysis on the findings and results obtained from the case study are presented on chapter five. Lastly, the sixth chapter details the main conclusions of the study, its contributions to academics and professionals, as well as limitation, and opportunities for future work.

Chapter 2 – Literature Review

This chapter introduces the central topics which are covered throughout this research. In addition to providing an overview of the current understanding of DevOps, ITSM, and PM, it also contains a summary on to what extent these concepts have been articulated in present literature.

2.1. DevOps

A summary on DevOps, its background, definitions, and key practices is provided in this section. Main motivations towards its adoption by organizations, as well as an outline of current challenges frequently discussed in literature, are also presented.

2.1.1. DevOps Background

The DevOps (Development and Operations) approach has its origins dating back to 2009, when the DevOpsDays conference was first held in Ghent, Belgium (Kim et al., 2016; Perera et al., 2017). It embodies a "change in IT culture" (Šćekić et al., 2018) in which high-speed software delivery (Virmani, 2015), improved collaboration (Perera et al., 2017) and the implementation of continuous testing, integration and feedback activities are emphasized (Soni, 2015). Its end goal is as far reaching as it is ambitious: To minimize the time between the initial concept and the "production ready application" (Soni, 2015); to maximize speed of delivery, having new code and software in "shippable state" at any point in time (Virmani, 2015); to capture, to the fullest extent, the benefits resulting from ongoing innovation (Sharma et al., 2015).

Many researchers have identified, analyzed, and documented the common existence of silo structures between development and operations teams (Rajkumar et al., 2016; Šćekić et al., 2018). As Wahaballa et al. (2015) noted, "In the IT industry, development and operations often are in conflict during deployment". DevOps practices and activities reportedly address said incompatibilities, enhancing the collaboration between development and operations workforce (Hüttermann, 2012; Guerriero et al., 2015) by way of shared experience and partnership (Rajkumar et al., 2016). Through ensuring team members responsible for operations are consulted regarding development topics, improved risk assessment (Gupta et al., 2019), better responses to market requirements (Bass et al., 2015), as well as quicker incident recognition and resolution are possible (Forsgren & Humble, 2015). In many cases, an overlap exists between teams and team members who have roles for software development, and those who are made responsible for its ongoing operational support (Cannon & Wheeldon, 2007).

DevOps plays a critical role for organizations in which efficiency, both at development and operational stages, is crucial for sustainability (Jessica Díaz et al., 2019; Paule et al., 2019); it is being described as a way of bringing the skills and knowledge of both teams closer (Senapathi et al., 2018), while keeping deployment and delivery costs low (Nybom et al., 2016).

High pressure to develop, package and release new code can be considered a hallmark of the current IT industry (Wahaballa et al., 2015), with companies such as Facebook implementing new software on a daily basis (Feitelson et al., 2013). The high-performance, high-availability and high-security requirements of contemporary Information Systems (IS) (Varga et al., 2019), coupled with the continuous engagement expected by present-day customers (Virmani, 2015), compels organizations to learn and apply new management strategies (Faustino, 2018) such as DevOps.

Internal motivating factors, such as waste reduction, improved collaboration, or better quality (Colomo-Palacios et al., 2018) are taken into account; external factors, such as customer expectations or adherence to market standards and best practices (Lwakatare et al., 2019) also play a role in the increasing volume of DevOps implementation projects.

2.1.2. Defining DevOps

DevOps has been the object of many definitions (Hemon et al., 2020). In fact, one of the main challenges organizations face when implementing its practices lies in the vague nature of its essence (Riungu-Kalliosaari et al., 2016). The lack of standard and clearly determined practices as well as a logical overview of how they should be implemented (de Feijter et al., 2018) leads to organizations pioneering their own implementation (Erich et al., 2014). While the purpose is clear, the concept itself still holds ambiguity (Smeds et al., 2015).

The present research does not aim to proport consensus on a given definition, yet we see value in presenting two prevalent interpretations.

DevOps is often defined as a software development methodology in which integration between operations teams and software developers takes place (Wahaballa et al., 2015). Given the necessity for faster, highly reliable releases, practices such as continuous deployment, continuous integration, monitoring, and performance management are paramount (Lwakatare, Kuvaja, et al., 2016). Under this conceptualization of DevOps, a “tight coupling” of development and operations roles takes place (Kuusinen et al., 2018), aimed at shorter patch release times, higher deployment frequencies and better software quality (Mohan et al., 2018).

DevOps has also been defined as a broader organizational approach, or a movement, aimed at integration and collaboration in the delivery process of software products or services (Díaz, Perez, et al., 2019). In some cases, this is extended to interactions not only between development and operations staff, but also with the end users themselves (Heistand et al., 2019). Here, Humble & Molesky (2011) devised four foundational principles for DevOps, commonly known under the Culture, Automation, Measurement, Sharing (CAMS) acronym: Culture, denoting the importance of shared responsibility and the pursuit of a high-trust, transparent organizational environment; Automation, meaning its usage throughout the entire development lifecycle; Measurement, pointing to the setting and management of objectives aimed at creating value; and Sharing, suggesting the importance of knowledge sharing across all participants and in all involved departments. This perspective of DevOps, emphasizing *fluidity* (Ebert et al., 2016) between traditionally isolated silos, has led to researchers concluding that DevOps is “more a cultural shift for IT than a process tools shift” (Colomo-Palacios et al., 2018).

2.1.3. DevOps Practices

Independent from how one would conceptualize DevOps, there are common, distinguishable practices associated with its implementation (Díaz et al., 2018). We have already discussed DevOps’ emphasis on collaboration, the bridging of development and operations activities, and the pursuit for shorter release cycles. In this section we will describe key practices contemplated in it to achieve said purposes.

Organizations implementing DevOps practices seek to continuously deliver new software, while maintaining reliability, quality, and compliance to security standards (Šćekić et al., 2018). This is made possible by establishing Agile planning principles in which small, cross-functional teams, having defined high-level objectives, collaborate frequently (using Scrum or Kanban tools) towards iterative development (Šmite et al., 2020). Said development cycles are commonly referred to as *sprints*. As pointed out by Beulen (2019), the successful implementation of these planning structures may require some “organizational rewiring”; a change in mindset should take place, transitioning from the concept of the big release, and equally large integration period, to the daily interaction, quick problem resolution and constant feedback (Caprarelli et al., 2020).

Establishing proper planning practices allows the introduction of the Continuous Integration (CI) concept in the coding and building of software. The implementation of a CI pipeline is considered an essential DevOps practice (Dyck et al., 2015) and a strong indicator of high IT performance (Debroy et al., 2018). CI refers to the early and continuous integration of code, enabling both faster feedback cycles and reduced workload associated with otherwise larger integrations (Caprarelli et al., 2020). The term itself was proposed much earlier by Booch et al. in 1991 stating, “in this evolutionary approach there is no big bang integration” (Booch, 1991). In effect, CI means that new development is committed to the main code either daily, or whenever an atomic objective is completed (Humble & Farley, 2010). Each small integration should automatically trigger a round of testing, easily identifying new issues and risks, resolving them early, and enabling increased deployment frequencies (Duvall et al., 2007).

With code being continuously developed and integrated, DevOps relies on automated deployment pipelines to release new software as frequently as possible (Kuusinen et al., 2018; Jiménez et al., 2019). The practices of Continuous Delivery (CDE) and Continuous Deployment (CD) govern the process by which new development is brought into production, adding value to end-users as soon as possible (Düllmann et al., 2018). Under CD, the end-to-end process of integration, testing and deployment is done in an automated manner; CDE on the other hand requires a manual intervention when the deployment to production is carried out.

The introduction of CD/CDE allows for application changes to quickly and reliably be moved from a “software repository to the customer’s hands” (Humble & Farley, 2010). Integrated and tested artifacts contained in deployment packages are analyzed, ensuring that all needed information, from space requirements, to installation processes and fallback plans, are in place for a successful delivery (Mohan et al., 2018). This practice has been characterized as the “heart of DevOps” and a critical element of software delivery optimization (Virmani, 2015).

Both CD and CDE can be considered extensions of CI, the main idea being that new software is always in a deployable state (Chen, 2017). Here, a bridge between development and operations is important: On one hand, developers ensure that reliable, high quality software is being built and released; on the other, operations staff deploy and review its behavior and monitor its performance in a production environment (Shahin et al., 2016). As such, collaboration between both teams should take place at every stage of the cycle (Gupta et al., 2019), relying on automation wherever possible (Schäfer et al., 2013).

A visual representation of a DevOps framework, demonstrating the consolidation of planning, CI, and CD/CDE practices, as well as the link of Development and Operations activity, can be found in Figure 1. A summarized description of the presented, core practices is shown on Table 1.

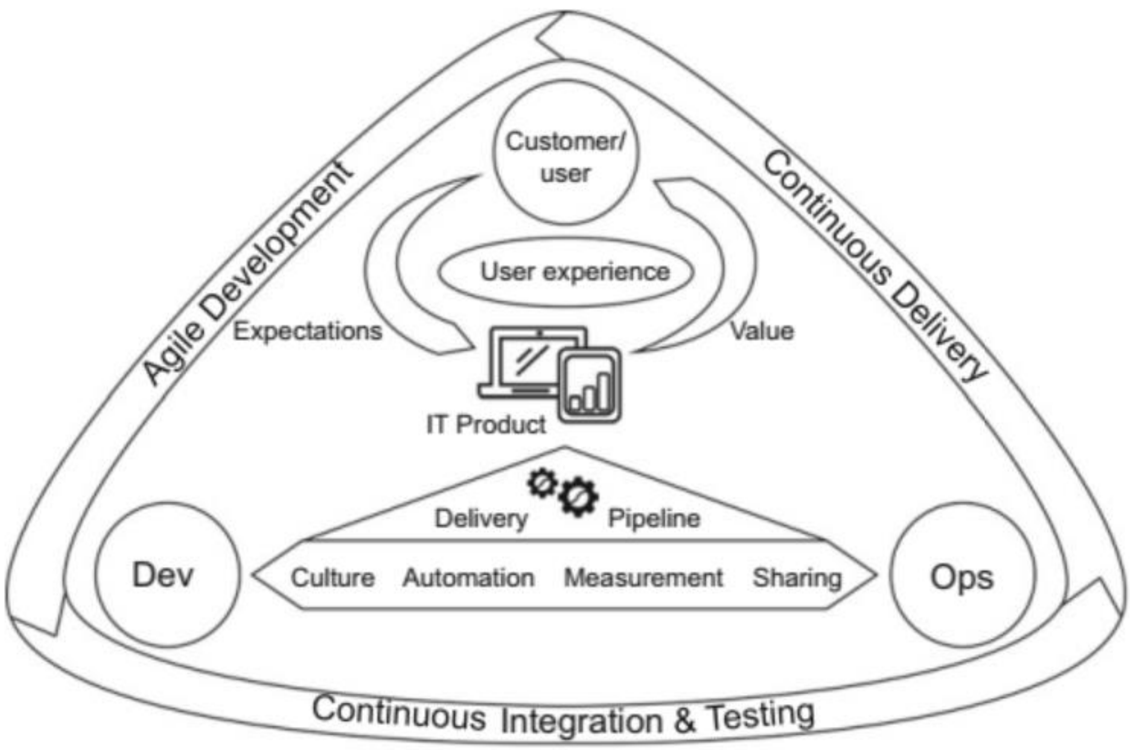

Figure 1 - A DevOps Framework (Alt et al., 2018)

Table 1 - Core DevOps Practices

Core Practice	Description
Agile Development	Implies frequent collaboration within small, cross functional teams aimed at iterative development. Usage of Scrum and Kanban tools in so-called *sprint* development cycles. (Šmite et al., 2020)
Continuous Integration	Consist in the continuous integration of new code into the main software repository whenever each defined objective is completed (Humble & Farley, 2010). This practice includes the triggering of (ideally automated) testing rounds to identify and resolve issues or mitigate risks (Duvall et al., 2007).
Continuous Delivery / Continuous Deployment	Deployment of new software into a production environment on a frequent basis (Kuusinen et al., 2018). The activity is either triggered manually (Continuous Delivery) or is entirely automated (Continuous Deployment). This is how value is delivered quickly and reliably to the customer (Virmani, 2015).

2.2. Information Technology Service Management

A summary on ITSM frameworks, their background and overall purpose, is presented in this section. Emphasis on ITIL is made, focusing on its relevance for businesses and how the PM process contained in it is conventionally executed.

2.2.1. ITSM Frameworks Background

The concept of IT service developed as organizations increased their reliance on technology to support business processes (Cannon & Wheeldon, 2007). This transformation, coupled with a global economic movement towards service-based economies, led to the rising adoption of ITSM frameworks (Pereira & Mira da Silva, 2012). These have become vital part of organizations (Mora et al., 2015), providing a set of processes to "align, design, deliver, manage and improve" both the internal use of IT resources (Wang et al., 2010), as well as the delivery of IT products to customers (Limanto et al., 2017). As such, ITSM has been considered an organizational imperative in the measurement of IT Service behavior (Faustino, Pereira & Mira da Silva, 2019).

As pointed out by Varga et al. (2019), a variety of models and frameworks for ITSM have been developed. These include COBIT, ISO / IEC 20000, ISO 9001 as well as CMMI (Sukmandhani et al., 2017); however, ITIL currently stands out as the most widely used ITSM framework (Sharifi et al., 2008; Cater-Steel et al., 2006) with its concepts being successfully adapted and adopted by a vast number of companies (Marrone & Kolbe, 2011).

ITIL was initially developed by the British government in the late 80's as a solution to growing service quality problems identified in IT sectors (*The Official Introduction to the ITIL Service Lifecycle*, 2007). The most recent ITIL release, ITIL4, began to be published in January 2019, building upon its predecessors and modernized with concepts from the increasingly prevalent Agile, DevOps and Lean methodologies (*What is ITIL | IT service management, AXELOS, 2019*). The overarching goal of the framework has prevailed: That of providing value, in the form of high-quality services, to organizations (*ITIL® Foundation Handbook 3rd ed. 2012*).

The overall result of a successful implementation of ITIL models has shown organizations achieving better performance in their key operational activities, with improved service quality and customer satisfaction as main outputs (Melendez et al., 2016). Reliance on an Incident Management (IM) process has proven to be an effective way of increasing service excellence and minimizing the negative impact of business interruptions (Yun et al., 2017). The introduction of a PM process, aimed at resolving the underlying cause of Incidents, has been observed as "one of the most important processes to ensure service stability" (Krishna Kaiser, 2018). To summarize, Service Management frameworks such as ITIL have been accounted to systematically provide better design, implementation, and continual improvement of IT operations (Abdelkebir et al., 2017).

Given the preponderance of the ITIL model amongst the diverse ITSM frameworks, we will be relying on its definition of PM throughout the present research. Due to the relative novelty of ITIL4, this research will be based on currently implemented practices and concepts as defined in ITIL 2011. However, to present how PM is defined under each ITIL iteration, Annex A was included, covering key definitions and process steps provided in both versions.

2.2.2. Problem Management Process

Based on the ITIL Foundation Handbook (UK, 2012), ITIL 2011 proposes an ITSM approach that emphasizes coordination and control throughout the various processes, roles, and systems existent in the end to end management of IT services. This structure is termed as the ITIL service lifecycle and consists of five stages: ITIL Service Strategy, ITIL Service Design, ITIL Service Transition, ITIL Service Operation and ITIL

Continual Service Improvement. The PM process, together with the Incident management, Event management, Request fulfilment and Access management processes, make up the ITIL Service Operation stage. A depiction of the ITIL service lifecycle can be found in Figure 2.

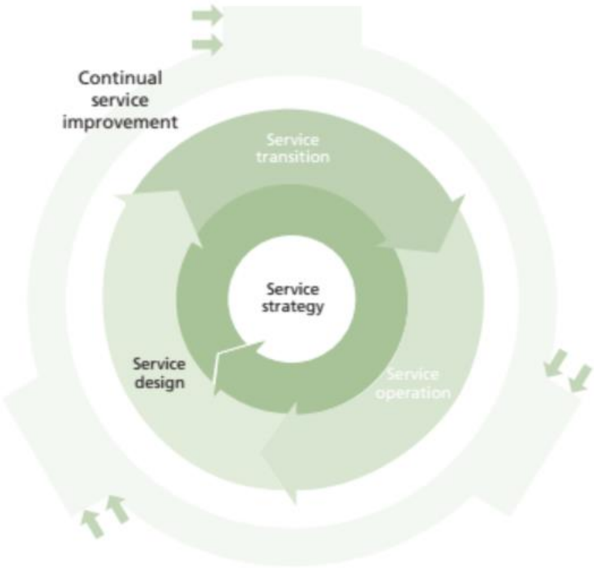

*Figure 2 - The ITIL Service Lifecycle (*ITIL® Foundation Handbook 3rd ed.)

According to the ITIL framework, PM is tasked with "managing the lifecycle of all Problem investigations" (*ITIL® Foundation Handbook 3rd ed. 2012*). In order to present the PM process, the concepts of Incident and Problem must be specified, as laid out in the ITIL library: An Incident is defined as an unplanned interruption or reduction in quality of an IT service; these can be triggered, for example, as a result of configuration errors or software bugs not identified in testing. Here, the Incident Management process is applied with the aim of restoring proper IT service behavior as quickly as possible. Conversely, a Problem is defined as *the underlying cause of one or more Incidents*. It is the purpose of PM, by way of monitoring, reporting and investigation, to discover, document and fix root causes of Incidents, preventing their reoccurrence (Sukmandhani et al., 2017).

Saarelainen & Jantti (2015) distinguish two types of Problem investigations: While Reactive PM is focused on directly addressing one or more Incidents, Proactive PM is aimed at "finding patterns and [incident] trends" that may suggest underlying faults in an IT service. Regardless of type, both are equally subject to Root Cause Analysis

(RCA) activities in which deficiencies are identified and improvement areas pointed out (Leszak et al., 2000). Common RCA techniques include Kepner-Tregoe analysis, the 5-Whys methodology, Ishikawa diagrams and the usage of Pareto charts.

According to the ITIL Foundation Handbook (2012), the PM process consists of six key activities: Problem detection; logging, categorization and prioritization; investigation and diagnosis; known error management; resolution; closure. Each key activity is described in Table 2. Outputs of an effective PM process include the sharing of knowledge, reducing Incident duration times by establishing Workarounds (Krishna Kaiser, 2018); furthermore, the implementation of Problem solutions, permanently removing underlying IT service errors, effectively mitigates the risk of future interruptions (Abdelkebir et al., 2017). In summary, PM is considered a crucial process towards providing reliable, high-quality IT services and maximizing customer satisfaction (Radhakrishnan et al., 2008; Sukmandhani et al., 2017).

Table 2 – Problem Management Process (ITIL® Foundation Handbook 3rd ed. (2012)

Problem Activity	**Description**
Problem Detection	Candidates for investigation are pinpointed as an output of Incident analysis.
Problem Logging, Categorization, and Prioritization	The Problem is recorded, described, and linked with associated Incident records.
Problem Investigation and Diagnosis	RCA activities take place, identifying the underlying cause of the Incident(s).
Known Error Management	Where possible, a Workaround, or temporary solution, is delivered to reduce the impact of Problems for which a complete solution is not yet implemented.
Problem Resolution	The implementation of solutions is carried out.
Problem Closure and Major Problem review, if applicable.	With the permanent solution in place, the investigation is formally finalized, with all findings and actions being documented. At the close of Major Problems, this should also be the stage used to define, document and implement lessons learned.

2.3. DevOps and ITSM

As mentioned in the introduction, very limited research has been done towards proposing and defining specific articulation between DevOps and ITSM frameworks such as ITIL (Kamuto & Langerman, 2017). Nevertheless, researchers such as Forsgren & Humble (2015) have recognized the potential relationship between both methodologies. Here, DevOps serves as the “the link between Software Development and Operations”, with Operations being systematized via ITSM frameworks (Hüttermann, 2012). In this chapter we will review some of the key considerations when discussing conceivable bridges between DevOps practices and ITSM frameworks.

The existence of siloed organizational structures is frequently pointed out as a barrier for the adoption of DevOps (Kuusinen et al., 2018). Having separate development and operations departments can bring about lack of collaboration, slowing down the release and deployment of software (Šćekić et al., 2018). In many organizations, not only are these divisions functionally separated, having their own work standards and procedures (Alt et al., 2018), they can also possess differing and potentially conflicting goals or incentives (Lwakatare et al., 2019). For example, on one hand we may find developers wanting to push a new release into production as soon as possible, while on the other hand we have operations personnel, seeking after a stable production environment, being motivated to slow down the deployment process (Kamuto & Langerhans, 2017; Humble & Farley, 2010). In this sense, Pilorget & Schell (2018) recognize the possibility of using DevOps inside an ITSM environment, stating that approaches which promote healthy collaboration between developers and operations are becoming essential for service-oriented IT organizations.

In Krishna Kaiser's (2018) proposal, a Problem Manager working under the ITIL PM process could have a direct link or presence in the planning sessions of a DevOps Team. In the event of a Problem being identified, it would be efficiently communicated to developers who can perform Root Cause Analysis. The introduction of permanent solutions would then follow via the DevOps flow of Continuous Integration, Testing and Deployment into production.

The interpretation of DevOps as an organization-wide approach, or as a cultural shift, gains substance as we discuss the improvement of relationships between departments. Most recent process optimization initiatives have focused on new software development practices, leaving “the operations side of software delivery lagging behind” (Virmani, 2015). Consequently, it is necessary for developers to understand the production environment and the ITSM framework that governs it (Hemon et al., 2020).

ITIL has been accounted by some as a source of rigidity, often conflicting with more agile development principles (Faustino, Pereira & da Silva, 2019). For example, in past ITIL literature, frequent releases were looked at as a source of poor quality and stability in the operations environment (*ITIL® Foundation Handbook 3rd ed. 2012*); the contrary is stated in Forsgren & Humble's (2015) analysis, where a proposition is made that greater throughput (based on deployment frequency and lead time to deliver) actually produces stability. Processes such as ITIL Incident, Problem and Change Management have a necessary role to play in the risk estimation, scheduling, monitoring and improvement of new releases (Lwakatare et al., 2019); as such, they have an impact on the rate by which new software is introduced (Šćekić et al., 2018).

The inclusion of operations personnel in development specific forums has been considered an important solution towards building efficiency and collaboration (Díaz et al., 2018). IT operators often lack in their understanding of application and system architecture (Bersani et al., 2016). By creating a cross-functional DevOps environment, "mixing development and operations people", concerns can be addressed, and approaches defined towards increasing cohesion and speed of delivery (Di Nitto et al., 2016). It can also be a way of integrating customer feedback, captured at the operations level, in the Continuous Delivery / Deployment process. By doing so, future releases are targeted at specific customer expectations and requirements (Soni, 2015). For this reason, researchers have stated that this should be done as early as possible, much before the end of a project (Bass et al., 2015). In some exceptional cases, the same team that is responsible for development of an application can also be the one overseeing its ongoing support (Cannon & Wheeldon, 2007).

The case for integrating development and operations practices has been made (Riungu-Kalliosaari et al., 2016; Nybom et al., 2016; Lucy Ellen Lwakatare et al., 2019). The study by Sćekić et al. (2018), in which DevOps was introduced for the development of a business intelligence software is a powerful example. Through cooperation between Ops and Dev, and by relying on continuous delivery, "there were no unexpected failures after setup changes" into production, with all errors being noticed and timely fixed. For this to regularly occur, a *blameless context* must exist (W. P. Luz et al., 2019); in other words, the focus should be on resolving problems, improving services, and sharing responsibilities. As Abdelkebir et al. (2017) put it, an agile ITSM integration is vital for improved quality and speed, as it is for superior user experience.

2.4. Related Work

This systematic analysis of related work was performed following a three-staged process of outlining, conducting, and reporting the review. We demonstrate this process in Figure 3.

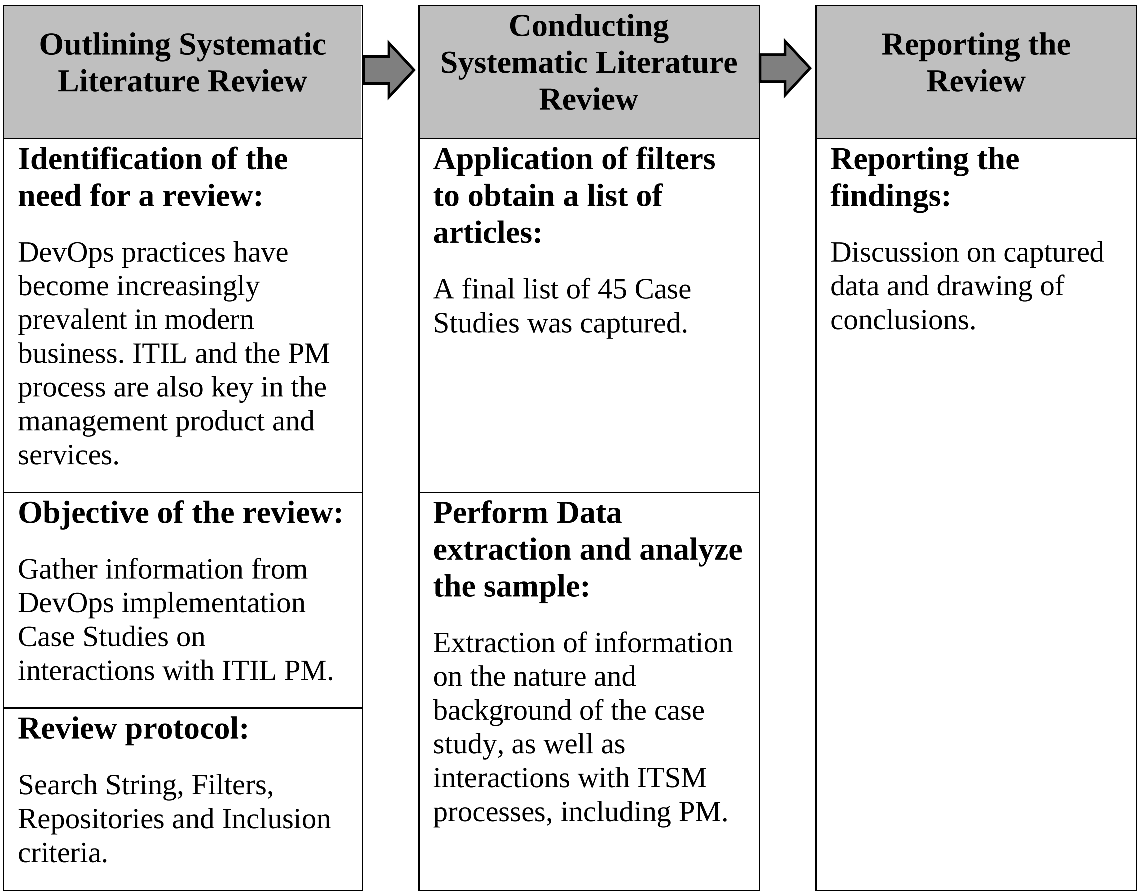

Figure 3 - Systematic Literature Review process

As was previously stated, search for literature specific to the role of DevOps practices in the application of ITIL processes yields little results. In order to present and integrate available related work, effort was done to capture DevOps implementation Case Studies; from these studies we looked for and interpreted instances where a relation to ITIL processes is discussed. Between August and October 2020, the following databases were analyzed:

- IEEE Xplore Digital Library (https://ieeexplore.ieee.org/Xplore/home.jsp)
- SpringerLink (https://link.springer.com/)
- ACM Digital Library (https://dl.acm.org/)

- SCOPUS (https://www.scopus.com/home.uri)
- ScienceDirect (https://www.sciencedirect.com/)
- EBSCO Academic Libraries (http://search.ebscohost.com/)
- Web of Science (https://www.webofknowledge.com/)

Our starting search string is based on the presence of the “DevOps” AND “Case Study” keywords. Afterwards, a variety of Inclusion criteria, Exclusion criteria and Filters were applied as can be seen on Tables 3 and 4. Results demonstrating the number of articles found on each database, as well as the number of Case Studies which remained after the application of Filters, is found on Table 5.

Table 3 - Inclusion and Exclusion Criteria

Inclusion Criteria	Exclusion Criteria
Written in English language	Not written in English language
Published from 2009 onwards	Published before 2009
Articles from Journals and Conference Proceedings	Non-scientific articles from Journals, Conference Proceedings and Books

Table 4 - SLR Filters

Filter ID	Filter Description
F.1	Abstract contains “DevOps” AND “Case Study"
F.2	Abstract indicated an implementation of DevOps
F.3	Duplicates are removed

Table 5 - Results and Filter Application

Databases	Keyword and Inclusion Criteria	F1	F2	F3
IEEE Xplore	28	25	13	13
Springer Link	39	10	10	10
ACM Digital Library	174	15	9	8
SCOPUS	118	81	27	8
Science Direct	116	5	3	1
EBSCO	24	17	4	1
WoS	52	43	19	4
Total	**551**	**196**	**85**	**45**

Each of the resulting studies was read for evidence of integration between DevOps practices and ITIL processes. Of the 45 case study articles that remained following the application of defined filters, nine were additionally excluded. Seven for not actually being Case Studies on the implementation of DevOps, one for being an older version of another study which was also captured in our search, and one for not focusing on the DevOps subject matter. Of the final list of 36 valid Case Studies, three explicitly mention a link between DevOps and ITIL. A list of these articles, their references, and vectors of analysis, are found on Table 6.

Referring to Table 6, in CS.7, Abdelkebir et al. (2017) propose and implement a DevOps inspired "Agile IT" framework in a large-scale organization. Upon setting out key ITIL processes and their activities, the authors present a model based on the Deming wheel, establishing a four-phase "quality approach for continuous IT improvement": Discover, Do, Act and Optimize (DDAO). The authors suggest that through the increased collaboration resulting from the application of DevOps practices, one can develop highly responsive and efficient ITSM capabilities.

Emphasis is placed on the use of "an efficient, agile and practical" ITSM approach to achieve organizational sustainability, service quality, user satisfaction and cost reduction. An "ITSM Agility Maturity Model based on DevOps" was also developed as way of evaluating results. The impact of DDAO, the usage of the model, or presented maturity assessment, is not specifically explored for each individual ITIL process. We have included the proposed "Agile IT" framework below, in Figure 4.

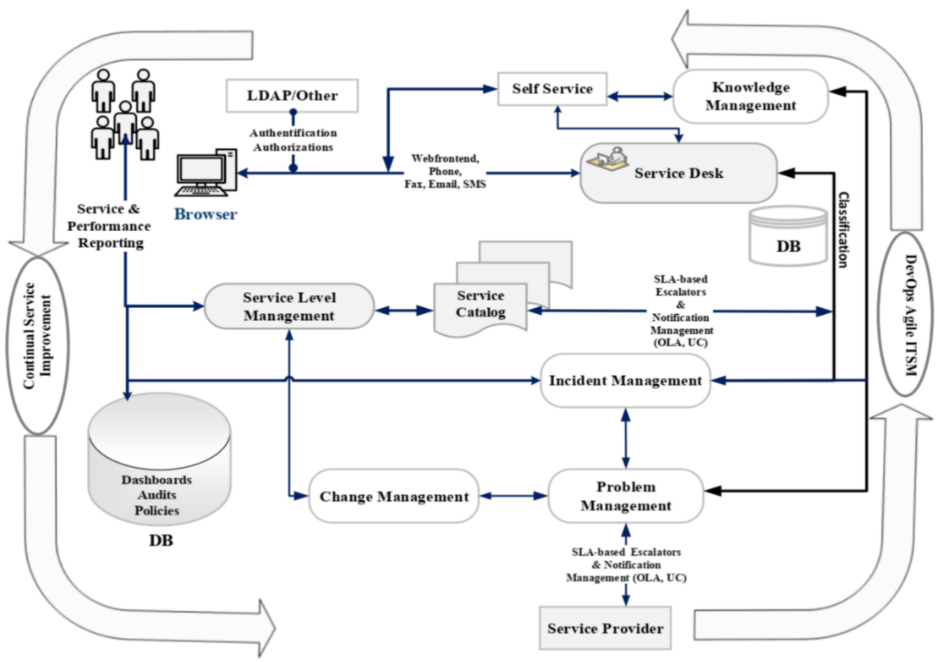

Figure 4 - Agile IT Framework (Abdelkebir, Maleh & Belaissaoui, 2017)

Table 6 - DevOps Implementation Case Studies

ID	Reference (author, year)	Location Country	Location Continent	Business Sector	ITSM Link
CS.1	(Šmite et al., 2020)	Sweden	Europe	Information Technology	No
CS.2	(Colomo-Palacios et al., 2018)	Spain	Europe	Human Resources	No
CS.3	(De Sanctis et al., 2020)	Italy	Europe	Lighting Business	No
CS.4	(Kuusinen et al., 2018)	Denmark	Europe	Information Technology	No
CS.5	(Di Nitto et al., 2016)	N/A*	N/A*	N/A*	N/A*
CS.6	(W. Luz et al., 2019)	Brazil	South America	Government Organization	No
CS.7	(Abdelkebir et al., 2017)	Morocco	Africa	Information Technology	**Yes**
CS.8	(Šćekić et al., 2018)	Montenegro	Europe	Banking Industry	No
CS.9	(Mohan et al., 2018)	Germany	Europe	Information Technology	No
CS.10	(W. Luz et al., 2018)	Brazil**	South America**	Government Organization**	No**
CS.11	(Schork et al., 2019)	N/A*	N/A*	N/A*	N/A*
CS.12	(Debroy et al., 2018)	USA	North America	Information Technology	No
CS.13	(Gupta et al., 2019)	Multinational	Multinational	Healthcare	No
CS.14	(Sampedro et al., 2018)	USA	North America	University Project	No
CS.15	(Riungu-Kalliosaari et al., 2016)	Finland	Europe	Information Technology	No
CS.16	(Senapathi et al., 2018)	New Zealand	Oceania	Finance & Insurance Industry	No
CS.17	(de Feijter et al., 2018)	N/A*	N/A*	N/A*	N/A*
CS.18	(Heistand et al., 2019)	USA	North America	Government Organization	No
CS.19	(Díaz, Perez, et al., 2019)	Spain	Europe	Information Technology	No
CS.20	(Díaz et al., 2018)	Multiple***	Multiple***	Multiple***	No
CS.21	(Lwakatare et al., 2019)	Finland	Europe	Information Technology	No
CS.22	(Ghantous & Gill, 2018)	Australia	Oceania	Information Technology	No
CS.23	(Smeds et al., 2015)	Finland	Europe	Information Technology	No
CS.24	(Jiménez et al., 2019)	USA	North America	Information Technology	No
CS.25	(Düllmann et al., 2018)	N/A*	N/A*	N/A*	N/A*
CS.26	(Caprarelli et al., 2020)	Multinational	Multinational	Information Technology	No

ID	Reference (author, year)	Location Country	Location Continent	Business Sector	ITSM Link
CS.27	(Hemon et al., 2020)	Multiple***	Europe	Information Technology	No
CS.28	(AL-Zahran & Fakieh, 2020)	Saudi Arabia	Asia	Mixed	No
CS.29	(Beulen, 2019)	(Not Provided)*	(Not Provided)*	(Not Provided)*	No*
CS.30	(Albuquerque & Cruz, 2019)	(Not Provided)	(Not Provided)	Finance & Insurance Industry	No
CS.31	(Hemon-Hildgen et al., 2019)	(Not Provided)	(Not Provided)	Information Technology	No
CS.32	(Wiedemann et al., 2019)	Multiple***	Multiple***	Multiple***	**Yes**
CS.33	(Gall & Pigni, 2018)	Multinational	Multinational	Information Technology	No
CS.34	(Jones et al., 2016)	UK	Europe	Information Technology	No
CS.35	(Nybom et al., 2016)	Multinational	Multinational	Information Technology	No
CS.36	(Furfaro et al., 2016)	N/A*	N/A*	N/A*	N/A*
CS.37	(Díaz, Pérez, et al., 2019)	Spain	Europe	University Project	No
CS.38	(Bruza, 2018)	USA	North America	Government Organization	No
CS.39	(Shahin et al., 2016)	Multinational	Multinational	Information Technology	No
CS.40	(Laukkanen et al., 2017)	Sweden	Europe	Information Technology	No
CS.41	(Pietrantuono et al., 2019)	(Not Provided)**	(Not Provided)**	(Not Provided)**	No**
CS.42	(Lwakatare, Karvonen, et al., 2016)	Finland	Europe	Information Technology	No
CS.43	(Alt et al., 2018)	Germany	Europe	Information Technology	**Yes**
CS.44	(Virmani, 2015)	(Not Provided)	(Not Provided)	Information Technology	No
CS.45	(Paule et al., 2019)	N/A*	N/A*	N/A*	N/A*

* These Case Studies were excluded from the analysis. Despite validation based on Search criteria, Inclusion Criteria and Filters, a full article reading led to the conclusion that these were not actual DevOps implementation Case Studies.

** These Case Studies were excluded for the analysis. CS. 10 is an older version of C.S 6; as such, the most recent article was used for the analysis. C.S 41 does not focus on DevOps but on another DevOps concept (DevOpsRET).

*** These Case Studies joined DevOps implementation insight from organizations operating in various locations and, occasionally, multiple business sectors. As such, we opted to register them as “Multiple”.

In CS.32, Wiedemann et al. (2019) very briefly mention ITIL as a set of processes employed by operations people to manage incidents in a structured manner. According to the authors, this is a feature of "traditional organized IT"; conversely, in cross-functional DevOps structures, each and every member is expected to be able to solve incidents, with the Team being ultimately responsible for the running of the service.

The work of Alt et al., (2018) registered as CS.43, has been concisely mentioned in our DevOps literature review. In this case study performed for a German IT company, an "Enhanced DevOps Framework" (Figure 1) is presented. ITIL is described as a key component of Service Management, and an essential element towards ensuring that DevOps practices such as CI and CD lead to expected business benefits. The authors specify that should there be issues in "adequately taking care of the requirements of operations for a smooth launch", negative user experience will follow in the form of errors and delay. As in CS.7 and CS.32, the authors do not delve deeper into the influence DevOps practices may have on any specific ITIL process.

Based on the outcomes of the SLR, the application of DevOps practices in a ITIL PM process was found to be a scarcely studied field. As such, the present exploratory research aims to provide further insight on the matter, contributing to the presently limited library on DevOps and ITIL integration.

Chapter 3 – Research Methodology

Exploratory research is aimed at examining observed phenomenon in which there is either an absence or scarcity of related works (Zainal, 2007). This is often done by way of case study, a valuable source from where researchers can both develop an “in-depth” understanding of complex issues in a “real life” setting (Dobson, 1999) or construct new theory (Perry et al., 2004). According to Thomas (2016), researchers should explore phenomenon having an intrinsic, instrumental, evaluative, explanatory, or exploratory motivating purpose; they should also define one among various types of case study: Special or outlier, to investigate out of the norm phenomenon; Key case, when studying a frequently occurring phenomenon or a Local knowledge case, where a researcher investigates a subject in which he has familiarity.

Given that the researcher works on a team that applies PM as well as an organization which has implemented DevOps, we classified this case study as a local knowledge case. Considering the early stage of research done on the subject of DevOps applicability toward the PM process, this investigation is exploratory in nature and purpose. Our approach is focused on building theory on the subject matter and, as what is studied only targets a single unit of analysis, this is considered a single-case approach (Aberdeen, 2013). Figure 5 presents our case study path, based on Thomas’ (2016) model.

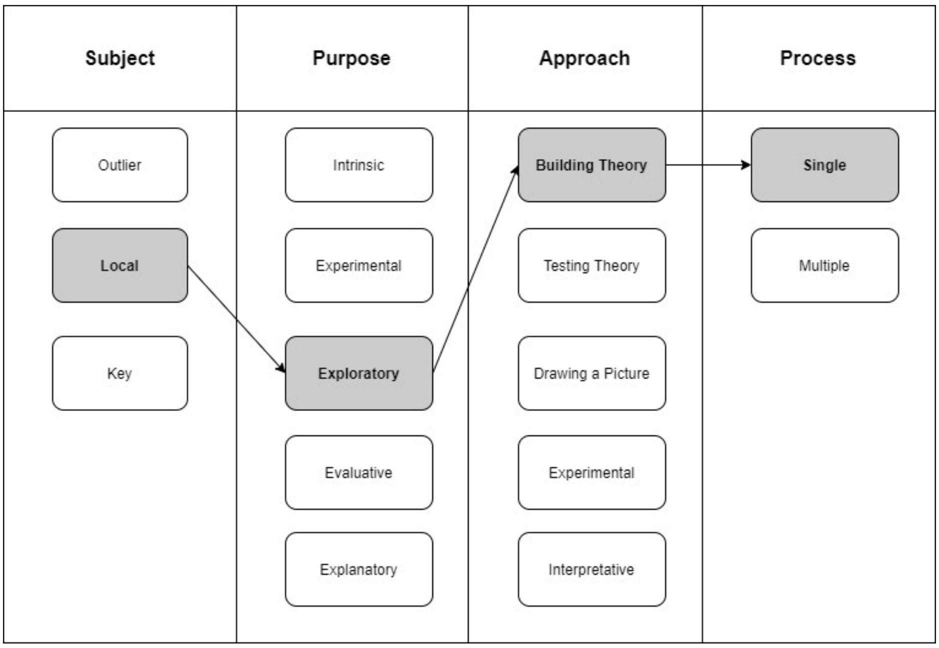

Figure 5 - Case study path, adapted from Thomas (2016)

A case study is built around a central question (Thomas, 2016). For the present research, that question is the following: "How can DevOps practices interface with an ITIL PM process?". The two central Research Questions (RQ's) used to shape and direct the design and course of our investigation (Tashakkori & Creswell, 2007) were identified and are presented in Table 7.

Table 7 - Research Questions

Research Question ID	Description
RQ1	What DevOps practices can be applied in each stage of the PM process lifecycle?
RQ2	Does DevOps improve the resolution of Problems? If so, how?

As this research relies on the experience and insight of a team operating under already implemented ITIL PM practices, this must be considered as a retrospective case study. As part of it, we relied on three data collection methods: Semi-structured interviews, process observation and document analysis, and a focus group exercise. These activities enabled the analysis of our issue through multiple angles, which is advisable in case study exercises (Modell, 2005).

The structure that was followed in the carrying out of this case study is based on Yin's (1994) recommendations as displayed in Table 8. Additionally, Figure 6 depicts the order of each selected data collection activity. In the following sub-sections additional detail regarding how the case study was designed and a demonstration of its validity are presented.

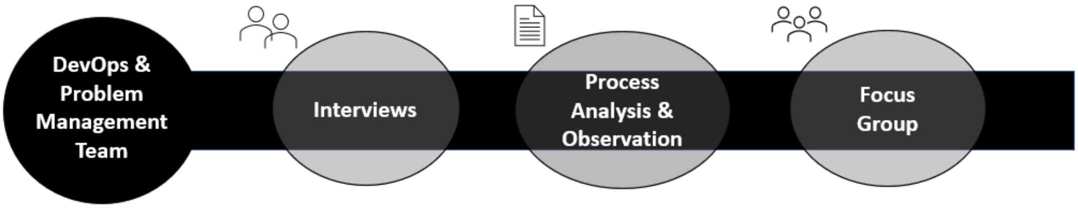

Figure 6 - Case study activities

Table 8 - Case study stages (adapted from Tellis, 1997)

Stage	Stage Description
Design case study Protocol	Composed of two stages: Determining the required skills and competences necessary for the carrying out of the case study; extensive reading on the topic in order to develop draft questions.
Conduct case study	Prepare and initiate data collection, leading interviews and defined focus groups.
Analyze case study Evidence	Evaluate gathered data by relying on an analytical strategy.
Develop Conclusions	Explain the benefits or challenges identified over the course of the exercise and present the conclusions captured as a result of data analysis.

3.1. Designing the Case Study Protocol

As presented in Table 8, the protocol designing stage of a case study is composed of two main activities. Firstly, an assessment is done to understand what competences are required for carrying out the case study; these can include ensuring that appropriate interpretation and listening abilities are in place, as pointed out by Tellis (1997). Secondly, an extensive reading of the subject matter is performed to enable the creation of draft questions and potential investigations routes.

For this research, it is essential that one leading the case study has knowledge and experience of both process management according to ITIL standards, with emphasis on the PM practice, as well as an understanding of DevOps practices in the IT realm. As a certified ITIL process owner with over 6 years of experience working in cross-functional teams, the author has the necessary competences to carry out and provide an interpretation of case study results.

A thorough literature review was also performed to deepen the author's understanding of PM and DevOps as applied in current business environments.

3.2. Conducting the Case Study

As previously mentioned, a triangulation approach was followed in the carrying out of this case study. It was initiated with the conducting of semi-structured interviews to capture practitioner experience and insight regarding the implementation of DevOps practices in a PM process. The format of said interviews was prepared in a way that enabled participants to freely share opinions and viewpoints on the subject matter, as recommended by Miles & Gilbert (2005). Additionally, a brief analysis of process data and workflows implemented by interviewed practitioners, as well as a focus group exercise, were also performed.

Selected interviewees are part of the IT department of a German multinational conglomerate company. More specifically, participants work in the process and operations management of the organization’s infrastructure department, providing assistance in the management of services and applications used by the company’s end users. This includes the overseeing of the internal PM process, as well as the management of the tools required by it and other implemented ITSM processes.

Further details regarding the participants are provided in Chapter 4, nevertheless it is important to mention the two different backgrounds encompassed in our case study. Out of 10 total interviewees, 6 oversee or participate in the PM process. This includes the carrying out of PM activities based on ITIL standards such as the detection and categorization of Problems, the carrying out of Root Cause Analysis, the leading of Problem resolution efforts and triggering of Problem Closure. The remaining 4 interviewees enable PM and other ITSM processes in the development and management of the tools required for service management. More specifically, these participants develop the ITSM platform used by the organization, via DevOps practices.

Analysis of the workflows being used, as well as the PM process in place, was performed. This allowed for an examination of conformity with the ITIL framework.

Based on the results from conducted interviews, potential bridges between DevOps and the PM process were identified and finally discussed in a focus group setting.

3.3. Analyzing the Case Study Evidences

The analysis of data collected from case study activities follows. In this stage, a review of the insight gathered from conducted semi-structured interviews, the PM and DevOps process observation, as well as the carried-out focus group exercise was done.

3.4. Developing Conclusions

In this stage a description of findings and conclusions taken from the captured and analyzed data is provided. Qualitative and quantitative information gathered from interviews, process analysis and observation, and the focus group exercise is summarized in the context of our Research Questions.

3.5. Case Study Validity

According to Yin (2009), four tests are recommended for the validation of case study reliability. A Construct Validity test refers to the "extent to which a study investigates what it claims to investigate" (Denzin and Lincoln, 1994); this implies the development of well-thought-out measures which mitigate subjective judgements in favor of evidence, even allowing a reader the ability to reconstruct the study himself (Yin, 1994). An Internal Validity, or "logical validity" test (Cook & Campbell, 1979), is applied to explanatory case studies in which causal relationships are explored; this test is not applied to an exploratory case study such as the present one, given that its aim is the building of new theory. An External Validity test is done to account for generalization; the understanding that theory should account for phenomena in various different settings (Calder et al., 1982). Lastly, a Reliability test aims to address the possibility of random error, ensuring that through transparency and replication, future researchers would reach the same conclusions, should they follow the same case study procedures (Denzin and Lincoln, 1994).

In Table 9 we present each test proposed by Yin (2009) as well as how and at which stage this exploratory case study passed each of them, making it a reliable case study.

Table 9 - Case study validation test outcomes (adapted from Yin, 2009)

Test	Purpose	Stage	Research Validation
Construct Validity	Development of measures which mitigate subjective judgements in favor of evidence.	Conducting the case study	A triangulation approach was followed in which data was captured from multiple sources including semi-structured interviews, process analysis and a focus group exercise.
Reliability	Mitigate random error through transparency and replication.	Designing the case study protocol & conducting the case study	Details regarding how this investigation was carried are provided so that in the future researchers may obtain similar results.
Internal Validity	Logical validity test to explore causal relationships.	Not applicable to exploratory case study	Not applicable to exploratory case study.
External Validity	Accounts for generalization. Theory should be applicable in various, different settings.	Designing the case study Protocol	A Literature Review was carried out in Chapter 2 in which no research was found specifically focused on the usage of DevOps practices in the PM process. The author notes this is evidence of the novelty of this research and a valuable contribution to the present body of knowledge.

Chapter 4 – Case Study Protocol and Conduct

Semi-structured interviews were carried out with 10 professionals working in or with previous experience on DevOps practices and the PM process. Each interview took on average 90 minutes, following a uniform and pre-defined set of questions, but allowing each interviewee time to expand upon given answers, expressing personal insight and opinion (Miles & Gilbert, 2005). Details on each interviewee are presented in Table 10.

Table 10 - Interviewee Details

Interviewee	Position	Experience in Role	Experience in IT	Experience in DevOps	Experience in PM	Number of Projects**
A	Process Manager	3	5	1	4	2
B	Developer	5	10	9	3	2
C	Service Manager	3	14	3	14	2
D	Developer	4	5	4	3	1
E	Process Manager	6	13	2	6	3
F	Developer Team Lead	4	12	12	12	4
G	Process Manager	6	10	1	6	1
H	Developer	4	12	4	4	3
I	Process Manager	3	10	3	3	4
J	Process Manager	1	6	1	4	2
Average		3,9	9,7	4	5,9	2,4

**experience is given in years.*
***by "number of projects" we mean, in how many organizations has the person applied PM / DevOps.*

Our set of interviewees is composed of 10 individuals with varying responsibilities in the organization but working in the same IT department. Save one exception, all have over 3 years of maturity in their current roles. Average participant IT experience is close to 10 years, with the majority being part of at least 2 previous projects.

Following the completion of semi-structured interviews, an analysis of processes being used in the organization was done, reviewing how PM and DevOps practices are being applied. This enabled both an assessment of the extent to which said processes are aligned with broader ITIL literature and provided additional context to findings captured from said interviews, through observation.

Lastly, a focus group exercise was carried out with 5 of the 10 interviewed professionals (Interviewees D, F, B, I and J) in which trends and findings resulting from the interviews, the process analysis and observation were discussed. This activity resulted in a more qualitative perspective on the outcomes of the case study.

Chapter 5 – Analysis of Case Study Results

This chapter presents the outcomes of the case study carried out, with an analysis of the results obtained from each of its three stages. The format, structure and product of semi-structured interviews, process analysis and observation, as well as the focus group exercise are documented.

5.1. Semi-Structured Interview Data Analysis

Conducted interviews were carried out following a standard three phase approach. Each began with basic questions regarding DevOps practices, gauging which were known to the interviewees, the level to which they were applied by them, the difficulty associated with their application as well as any notable benefits resulting from implementation. A second stage dedicated to the PM process followed. Here, questions were asked about the PM lifecycle, documenting which activities were known to the participants as well as which were directly applied by them. Lastly, in the final stage of the interview, participants were asked to assess the relevance of each known DevOps practice in each PM lifecycle activity.

The following subchapters provide detail on the results of the exchanges, their main findings, as well as relevant patterns observed in each stage of the interviews. It is relevant to point out that although DevOps knowledge and experience varied, all interviewees shared a solid and common understanding of the PM process as defined in ITIL.

5.1.1. DevOps

This stage of the interview was initiated by asking interviewees for their definition of what DevOps is. Answers to this introductory question are presented in Table 11. Afterwards, participants were shown a list of 12 DevOps practices and asked to firstly point out which ones they had knowledge of, and secondly, which ones have they applied. For the second question, regarding to what extent was the DevOps practice applied, the choice of three possible answers was presented. Either participants *did not apply* said practice, *partially applied* it, *or fully applied* it. Partial application refers to cases in which a practice was incompletely implemented, or not implemented to the extent the participant expected it to be, based on his understanding of it. Results of this question are presented in Table 12.

Overall, interviewees demonstrated a general understanding of what DevOps is. The differences in participant background seem to have played a role in the maturity of the answers given to our first two questions.

Table 11 - Interviewee DevOps Definitions

Interviewee	Question: *Please describe DevOps in your own words.*
A	"I don't know much about DevOps. I understand it is the joining of Development and Operations, which implies people working together and doing both: Applying new development and monitoring it live, through Operations. It ensures that there is warranty and utility in the services being provided."
B	"DevOps is a frame of mind used to approach development issues by taking advantage of Dev and Ops. This is done by keeping open communication, particularly in the resolution of issues. (…) It is about getting teams to work together. Collaboration is key. It is not about getting Operations to do Development work, or vice-versa."
C	"Never really used it fully. We use pieces of it during our day-to-day activities. Its goal is to have a more agile approach to the release of new applications among other things. It speeds up implementation."
D	"A multi-task team that has various types of skills and can adapt to different circumstances in a project. It is a team that needs to work together towards accomplishing goals. It is also a mindset: What is behind the collaboration and how each individual works in the Team."
E	"DevOps is a set of best practices that are aligned to deliver good software and applications to a customer."
F	"I would define it as a methodology that allows the delivery of a service, or a product, in a way that development and operations can collaborate on a continuous basis to increment that product or service."
G	"I have a basic understanding of DevOps. I see it as a means toward improved waste reduction, faster delivery, better collaboration, and silo reduction."
H	"I see it as a way of overcoming a wall between Dev and Ops. It is a set of practices that dilutes the difference between people (...) DevOps takes the wall down between Dev and Ops through closer collaboration. It should be organic and come out of people's own initiative, when applicable."
I	"A combination of Development teams and Operations. Having both together allows for practical and technical knowledge to resolve issues and improve IT services."
J	"It is basically an integration between Development and Operations; a way to shorten or straighten relations between both. Instead of long dev cycles and large deployments with even longer times between those deployments, the idea is to have this done in smaller chunks which are frequently deployed. This gives a better ability to responded to business needs, hopes, and customers. It is highly focused on the minimal viable product; getting something that works with minimal time for testing."

Those currently working on Developer roles (interviewees B, D, F and H) presented a greater degree of confidence in their responses, possibly related to their more direct and practical involvement in DevOps practices comparatively to Process and Service Managers.

Of the 12 practices, *Dev & Ops Feedback Loops* and *Automated Monitoring* were known by all participants; *Infrastructure as Code* was known by a single participant but applied by none. *Stakeholder Participation*, *Process Standardization*, *Continuous Planning* and *Dev & Ops Feedback Loops* were identified as the most widely applied practices.

Table 12 - DevOps Practices Known & Implemented

	Shift-Left	Continuous Planning	Dev & Ops Feedback Loops	Continuous Integration	Automated Monitoring	Application Prototyping	Deployment Automation	Test Automation	Infrastructure as Code	Stakeholder Participation	Process Standardization	Change Management	Total	Percentage
DevOps Practices Known														
A	●	●	●	●	●		●			●	●	●	9	75%
B		●	●	●	●	●	●	●			●	●	9	75%
C	●	●	●		●		●	●		●	●	●	9	75%
D	●	●	●	●	●	●	●	●		●	●	●	11	92%
E		●	●	●	●		●	●	●	●		●	9	75%
F		●	●	●	●	●	●	●		●			8	67%
G		●	●	●	●			●		●	●	●	8	67%
H		●	●		●	●	●	●			●	●	8	67%
I			●	●	●					●	●	●	6	50%
J	●	●	●	●	●	●	●	●		●	●	●	11	92%
Total	4	9	10	8	10	5	8	8	1	8	8	9		
DevOps Practices Applied														
A	●			◖						●	●	◖	5	42%
B		◖	◖	◖	◖	◖	●	◖			◖	●	9	75%
C	●	◖	●		◖					◖	●	●	7	58%
D	●	◖	●	◖	●	●		◖		●	●		9	75%
E		●	◖	◖	●					●		●	6	50%
F		◖	◖	◖	◖	●		◖		◖			7	58%
G		●	◖	◖	●					●		◖	6	50%
H		●	◖			◖	◖	◖		◖	◖		7	58%
I		◖	◖							◖	◖	◖	5	42%
J	◖		◖	◖		◖				●	●	◖	7	58%
Total	3,5	5,5	5,5	3,5	4,5	3,5	1,5	2	0	7	5,5	5		

Legend: ● - Known / Fully Applied; ◖ - Partially Applied; - Not Known / Not Applied

Having discussed each interviewees' level of understanding over each DevOps practice, questions were asked regarding the challenges associated with their implementation, as well as the perceived benefits that resulted from them.

Table 13 presents the outcome of requesting each interviewee to rate the level of difficulty experienced when implementing each practice on a scale of 1, meaning *Very Hard*, to 5, meaning *Very Easy*. Note that the number of inputs to each practice depends on the number of participants that replied with having experience in either partially or fully applying said practice. Some comments were also added, as we allowed participants to justify their answers.

Similarly, Table 14 shows the outcome of requesting each interviewee to rate the extent to which benefits were experienced when and after implementing each practice in a scale of 1, meaning *No benefits*, 2, meaning *Neutral*, and 3, meaning *Observable benefits*.

We detected some lack of consensus when discussing the challenges associated with implementing various practices; interviewees frequently answered with a wide variety of viewpoints for how easy or how difficult a certain DevOps practice was to have implemented (e.g., *Dev & Ops Feedback Loops, Stakeholder Participation*). However, a more uniform view can be seen when discussing whether or not those practices have led to observable business benefits; here, there was a clear tendency towards positive answers across the board (e.g., *Shift Left*, *Continuous Planning*).

According to interviewee ratings, when reviewing the average difficulty associated with implementing each DevOps practice, *Deployment Automation* stands out with a 4,5 average, meaning and Easy to Very Easy implementation; it is important to note however that this particular result is based solely on the insight of two participants. If we study the three most widely implemented practices based on Table 12, we see a Neutral to Easy rating for *Continuous Planning* and *Dev & Ops Feedback Loops*, and a Neutral to Hard rating for *Process Standardization*. On the lower end of the scale, *Automated Monitoring* received the lowest score we captured, based on the insight of six interviewees, with three indicating a Very Hard implementation for this DevOps practice.

Table 13 - Challenge to Implement DevOps Practice

Practices	**Rates**						**Comments**
	1	**2**	**3**	**4**	**5**	**Average**	
Shift-Left		3		1		2,5	"Technically easy, but a lot of logistical implementation work."
Continuous Planning		3	1	2	2	3,4	"We recognized the need for it, and so everyone made the decision to embrace it. It was easy."
Dev & Ops Feedback Loops	1	2	3	1	2	3,1	"It is easier when we are working on the same goal; if so, people have interest in it."
Continuous Integration	1	2	1	3		2,9	"A very good design of how things are done absolutely needs to be in place."
Automated Monitoring	3	1	1		1	2,2	"The difficulty is in defining perfectly what the ideal state is, and then using automated monitoring to spot deviation. It is very hard to implement."
Application Prototyping			2	2	1	3,8	"The setting of clear expectations on the customer end is the more challenging part of this practice."
Deployment Automation				1	1	4,5	"The implementation of scripts to automate this part of the work is not very challenging with the tools that we have."
Test Automation		3			1	2,8	"Proper testing requires ample information from all sources. Without needed information and resources, this is very hard to do."
Infrastructure as Code						0	N/A
Stakeholder Participation	2	1	3	3		2,8	"Stakeholders generally want to be involved in what is being done. Participation on its own is not challenging. It is required to determine what forums to use."
Process Standardization	1	2	3		1	2,7	"Depends on the willingness of the organization to adopt standardized processes."
Change Management		2	2	3		3,1	"Difficult to implemented at the start, but gradually becomes neutral or easy as maturity increases."

Legend: 1 - Very Hard; 2 - Hard; 3 - Neutral; 4 - Easy; 5 - Very Easy.

Table 14 - Benefit from Implementing DevOps Practice

Practices	Rates				Comments
	1	2	3	Average	
Shift-Left			4	3	"Some tasks end up being done earlier in the lifecycle. This allows to free up time from more technical resources, reducing waste and quickening the process."
Continuous Planning			8	3	"Ensures that we are all constantly on the same page. It allows for us to be Lean, constantly thinking about next steps and priorities. Allows for flexibility."
Dev & Ops Feedback Loops		2	7	2,8	"There is a gap between Ops and Dev. Having this in place does require an investment from both ends, where we help each other. In a utopia, we are basically doubling teams."
Continuous Integration	1	1	5	2,6	"The way developers work for improving our ITSM tool is through continuous integration: Implementing small changes and features constantly. It allows for us to have quick corrections implemented in a few hours / minutes."
Automated Monitoring		1	5	2,8	"We are able to see the status of things without human intervention."
Application Prototyping			5	3	"From a Dev side we benefited. It is however very important that expectations of what the prototype should be are clearly defined."
Deployment Automation	1	1		1,5	"Ours [referring to the ITSM tool implemented] helps with bulk changes, reducing some manual work, but the effort is never entirely automated."
Test Automation		2	2	2,5	"These "sanity checks" have been applied with success. A dedicated team is in place for it specifically."
Infrastructure as Code				0	N/A
Stakeholder Participation	1		8	2,8	"Having [stakeholder] visibility and participation in the work adds to the reputation of the Team, which can improve performance."
Process Standardization	1	1	5	2,6	"There is a standard scrum process in place, but it can change depending on how day to day activities are done. Some flexibility is still needed for motivation, however."
Change Management	1	1	5	2,6	"Makes everyone aware of what will change ahead, knowing the standard practices associated with the process."

Legend: 1 – No benefit; 2 - Neutral; 3 – Observed benefit.

Analyzing the outcomes of our question regarding the business benefits observed from each DevOps practice, we find unison agreement on the positive impact of *Shift-Left*, *Continuous Planning* and *Application Prototyping*. If we once again review the top three most widely implemented practices based on Table 12, we see overall positive responses for all three. The only practice which was discussed as having a neutral to no observable benefit was *Deployment Automation*, based on the responses of two participants.

The *Infrastructure as Code* practice was not taken into consideration for the above analysis given that none of the ten interviewees had experience applying it.

5.1.2. Problem Management

Having completed a first DevOps focused section of the interview, the author continued by asking interviewees questions regarding their understanding of the PM process, the PM lifecycle, and the extent to which participants had applied each practice of the lifecycle.

Answers to our introductory question on the definition of PM are presented in Table 15. Results to the question on which practices in the PM lifecycle participants knew, and the extent to which they were applied, are shown in Table 16. As done for Table 12, the choice of three possible answers was presented. Either participants *did not apply* each practice, *partially applied* it, or *fully applied* it.

Interviewees demonstrated good understanding of what PM is, its purposes and structure, according to the ITIL framework. Contrary to what was noted in the DevOps section, where Developers were more familiar with DevOps concepts and practices over Process and Service Managers, mature knowledge of the PM process was observed in participants regardless of their current roles in the organization; this may have been a consequence of the interviewed Developers having previous work experience performing various Process Management activities.

All interviewees answered positively in regard to their knowledge of each of the 6 PM practices. When asked about the extent to which each has applied them, we point out *Problem Resolution*, *Problem Detection* and *Problem Logging* as the more widely implemented practices. *Known Error Management* was the lesser implemented practice, despite all interviewees stating that they have implemented it to some extent; from reviewing the answers we find that only 3 participants fully applied it.

Table 15 - Interviewee Problem Management Definitions

Interviewee	**Question:** ***Please describe PM in your own words.***
A	"The process used to tackle complex and widespread issues. We try to find the cause of issues, resolve them permanently, or at least minimize their impact should they happen again. It is a role of our investigations to find the cause of things and identify, through planning and observation, how impact can be mitigated."
B	"People log Incidents, these can sometimes be identified as Problems. If so, Root cause analysis takes place, and the investigation goes through phases aimed at coming up with a solution and raise a Change to fix the issue. PM is the process that manages the lifecycle of Problems. It ensures success with fix implementation by making it so that there are no reoccurrences."
C	"A process that helps us address root causes, or underlying causes, of one or more Incidents. At the end, it is a process which exists to address and eliminate issues, decreasing impact on the end users and improving service quality."
D	"A process whose main goal is to solve issues which are not solvable solely under Incident Management. Problems are larger than single Incidents, and they require permanent solutions."
E	"PM is all about learning from mistakes. It is about figuring out what can be improved and what can be done in a better way to prevent future incidents."
F	"PM is a process or practice that takes care of managing the lifecycle of all Problems. Problems are the underlying cause of one or more Incidents. Resolving Problems mitigates incidents and leads to service improvement, reduced risks, better reliability, and less Incidents."
G	"The process used to avoid incidents, to satisfy the customers, to ensure services are ongoing and not failing. It should be proactive, in the sense that it is proactively avoiding future incidents."
H	"This is the process to address reoccurring Incidents. Without it there is no investigation, there is no documentation, there is no improvement. This is an extremely important process, and it tells us a lot about the company's maturity: Where this is used, there is care in figuring out the roots of issues."
I	"Everyone is looking for a permanent solution to their problems, and that is what PM is aimed at: For the cases where a permanent solution is needed. It is the proper process to provide a stable solution for issues, preventing recurrent incidents."
J	"PM is the process that ensures there is something in place to standardize the approach towards Root Cause Analysis, its review, the tracking of fixes and mitigation activities, and the resolution of causes for Incidents. It is the link between technical investigation, documentation as well as the steps and follow-up actions towards resolution."

Table 16 – Problem Management Practices Known & Implemented

	Problem Detection	**Problem Logging**	**Problem Investigation**	**Known Error Management**	**Problem Resolution**	**Problem Closure**	**Total**	**Percentage**
PM Practices Known								
A	●	●	●	●	●	●	6	100%
B	●	●	●	●	●	●	6	100%
C	●	●	●	●	●	●	6	100%
D	●	●	●	●	●	●	6	100%
E	●	●	●	●	●	●	6	100%
F	●	●	●	●	●	●	6	100%
G	●	●	●	●	●	●	6	100%
H	●	●	●	●	●	●	6	100%
I	●	●	●	●	●	●	6	100%
J	●	●	●	●	●	●	6	100%
Total	10	10	10	10	10	10		
PM Practices Applied								
A	●	●	●	◖	●	●	6	100%
B	◖	●	◖	◖	◖		5	83%
C	●	●	◖	●	●	●	6	100%
D	◖	◖	●	◖	◖	◖	6	100%
E	●	●	●	◖	●	●	6	100%
F	◖		◖	◖	●	◖	5	83%
G	●	●	◖	●	●	●	6	100%
H	●	●	●	●	●	●	6	100%
I	●	●	●	◖	●	●	6	100%
J	●	●	●	◖	●	●	6	100%
Total	8,5	8,5	8	6,5	9	8		

Legend: ● - Known / Fully Applied; ◖ - Partially Applied; - Not Known / Not Applied

5.1.3. DevOps and Problem Management

Having covered DevOps, its understanding, practices, challenges, and benefits, as well as PM, its knowledge and implementation, the author proceeded to join both areas together by asking interviewees to rate the relevance each DevOps practice has in each PM lifecycle practice. Participants were asked to indicate this by selecting one of the following options: 1, meaning *low relevance*; 2, meaning *relevant*; and 3, meaning *high relevance*. Table 17 presents the outcome of this exercise. In it, a compilation of answers each interviewee provided was carried out, adding the values entered for each cell, and grading the result using three colors. Light grey indicating slight significance, where the added total of answers was between 10 and 14; dark grey indicating significance, where the added total of answers was between 15 and 19; and black, indicating high significance, with the added total of answers equaling to or being over 20. Where applicable, it includes opinions and justifications provided by interviewees as they responded to this question (these being represented as Ix's). Said comments are found on Table 18.

Table 17 - DevOps Practice significance in the Problem Management lifecycle

		PM Practices					
		Problem Detection	Problem Logging	Problem Investigation and Diagnosis	Known Error Management	Problem Resolution	Problem Closure
DevOps Practices	Shift-Left	8	6	6	10 **(F1)**	6	6
	Continuous Planning	14 **(F2)**	12	16 **(F3)**	17	20 **(F4)**	12
	Dev & Ops Feedback Loops	22 **(F5)**	16	21 **(F6)**	17 **(F7)**	19 **(F8)**	17
	Continuous Integration	9	10	12 **(F9)**	12	16 **(F10)**	7
	Automated Monitoring	24 **(F11)**	12 **(F12)**	17	12	13	12 **(F13)**
	Application Prototyping	6	7	8	7	9 **(F14)**	6
	Deployment Automation	5	4	4	4	5	5
	Test Automation	8	7	8	6	9 **(F15)**	6
	Infrastructure as Code	0	0	0	0	0	0
	Stakeholder Participation	21 **(F16)**	16 **(F17)**	14	16	19 **(F18)**	16
	Process Standardization	9	11	13 **(F19)**	14 **(F20)**	12	12 **(F21)**
	Change Management	13	10 **(F22)**	12	11	17 **(F23)**	10 **(F24)**

Legend: Slightly significant; Significant; Very significant.

Both Table 17 and Table 18 are key inputs towards answering RQ1, providing us insight on which DevOps practices would be applicable for the various phases of the PM lifecycle.

Table 18 - Insight on DevOps & Problem Management Practice matches

Feedback	DevOps Practice	PM Practice	Comment
F1	Shift Left	Known Error Management	"When a proper Known Error Database is implemented, there are plenty of opportunities for shift-left (…) Known Error workarounds can often be applied at the 1st level of support." "This is a key element of the link between the PM and Knowledge Management process (…) Having this set of practices in place is an enabler to proper Knowledge Management."
F2	Continuous Planning	Problem Detection	"Having a regular forum where potential Problem candidates are discussed can be helpful to the process." "With continuous planning we can continually find issues and obstacles to be addressed via the PM process."
F3		Problem Investigation and Diagnosis	"This would be an important practice to have as it enables better task management (…) knowing where each investigation is on an ongoing manner is helpful for the process."
F4		Problem Resolution	"Solution activities for Problems have to be planned. Having [Continuous Planning] is very relevant as it can expedite the implementation." "Continuous planning sessions, the fact that they allow the opportunity for ongoing discussion, can be a positive element for the process." "This can be used to also prevent Problems even before they get to Production."
F5	Dev & Ops Feedback Loops	Problem Detection	"Being aware of what development is to be implemented, and pointing out risks or issues, is an important contributor to the creation of Problems."
F6		Problem Investigation and Diagnosis	"Performing Problem Root Cause Analysis always requires collaboration. Developers and Operations need to work together when it comes to figuring out the cause of Problems." "Regardless of who actually does the Root Cause Analysis, it is important to have consistent feedback between us process teams and the tech teams. It is how we obtain necessary updates and improve communication."
F7		Known Error Management	"The validation of Known Errors and Workarounds has to be checked and confirmed by the Development side, who often have the technical awareness to approve or reject this."
F8		Problem Resolution	"This combination is needed to make the requirements of a Problem solution clear and align on how that solution will be introduced properly." "It is the Developers who take the lead in the working out of a resolution for a Problem; however, the implementation of it already involves Operations. They should be working together."
F9	Continuous Integration	Problem Investigation and Diagnosis	"A Problem investigation is an ongoing process. It should be easy to track and simple to manage. The idea of Continuous Integration, with new information being added to the Problem piece-by-piece, makes sense."
F10		Problem Resolution	"In the implementation of solutions for Problems, this practice can speed thing up to a higher pace." "This [Continuous Integration] is how we should aim to implement Problem actions."

Feedback	DevOps Practice	PM Practice	Comment
F11	Automated Monitoring	Problem Detection	"This is a key practice to proactively identify Problems. We need to be aware of what is happening in the environment; having automation helps." "We could link this with the Event management process, working as an input to PM."
F12		Problem Logging	"There may be opportunities to automate the creation and logging of Problem records based on certain monitoring triggers."
F13		Problem Closure	"We could use some sort of automated monitoring to actually confirm the complete resolution of a Problem investigation."
F14	Application Prototyping	Problem Resolution	"The prototyping of a Problem solution could be done."
F15	Test Automation	Problem Resolution	"We could use this practice to test how effective a Problem solution is before we implement it in Prod."
F16	Stakeholder Participation	Problem Detection	"Additional 'eyes on the field' are important to detect things as soon as something goes wrong."
F17		Problem Logging	"The prioritization of a Problem and its classification, based on urgency and impact, is dependent on the Stakeholders insight and participation."
F18		Problem Resolution	"Stakeholder should be involved in the confirming of solutions to the Problems" "They are the best suited to consider, agree, comment on and confirm the solutions to Problem issues."
F19	Process Standardization	Problem Investigation and Diagnosis	"Having standard processes allows us to organize and help the carrying out of investigations. Standardization also more easily points out what may have failed during a Problem."
F20		Known Error Management	"Only with a standardized process within our teams can we ensure that a good Known Error Knowledge Base is in place; it prevents wasted time where we have people investigating matters that are already known or under resolution."
F21		Problem Closure	"In Closure, everything should be clearly documented, and everyone should be aware of and follow the same process for it. (…) the outputs of each activity towards Closure should have a predictable outcome.
F22	Change Management	Problem Logging	"We could trigger the automatic creation of Problems from Changes that are considered 'failed'."
F23		Problem Resolution	"Solutions implemented via PM should also go through a Change Management process."
F24		Problem Closure	"The outcome of the Change can be used as a confirmation that the issue was actually resolved. It is evidence that the Problem is ready for Closure."

5.2. Process Analysis and Observation

A review of the PM process being applied in the organization by its teams was conducted. Our objective consisted in analyzing its main activities to understand how aligned these are with the broader ITIL PM literature, observe if and what DevOps practices are used over the course of Problem investigations, and how they may influence the results of the process.

According to the PM process applied in the organization, and in line with ITIL literature, a Problem is defined as the "underlying cause of one or more Incidents or potential incidents".

Two types of Problem investigations are referred to: Reactive Problems, being those originating exclusively following a Major Incident, and Proactive Problems, being those encompassing all other possible sources that may trigger Problem investigations; this includes the outputs of system monitoring tools, results of Incident trend analysis, insight from Service Management teams and feedback from customers.

Each Problem investigation regardless of source, priority, categorization or type, progresses through a standard four-stage lifecycle. A summary of the applied PM lifecycle, and its alignment with standard ITIL PM activities, is found on Table 19. A flowchart depicting the 4 key activities of the lifecycle is presented in Figure 7.

Teams who have been granted accesses to the PM module of the ITSM tool being utilized are able to create new Problem candidates based on detected incidents, events and knowledge of existing faults in the business' IT landscape. Said teams tend to be comprised of Service Managers who collaborate closely with their service provider counterparts and other stakeholders from the organization.

Templates containing standard questions and requirements are used to ensure that the necessary information for the progressing of an investigation is documented. Meetings are also regularly scheduled to both confirm that validation is granted for each Problem, and to reach agreements on ownership for which service provider(s) will be carrying out Root Cause Analysis.

Table 19 - Comparison of ITIL & Organizations' Problem Management lifecycle

Standard ITIL PM Activities	Org. PM Stages	Objectives and outputs of each stage
Problem Detection	Problem Creation & Validation	Problem candidates are raised following either Major Incidents or based on the result of monitoring patterns, incident analysis and service management insight. Each candidate is reviewed for validation, ensuring it has the proper categorization, prioritization and estimated business benefits before an actual investigation is initiated.
Problem Logging, Categorization, and Prioritization		
Problem Investigation and Diagnosis	RCA Creation and RCA Review	Providers review the issues reported in the Problem record and provide both Root Cause Analysis (RCA) and proposals for solution activities that may be completed towards resolving the Problem. A Known Error may be generated upon delivery of the RCA Service Management reviews the delivered information and provides an approval or rejection.
Known Error Management		
Problem Resolution	Problem Resolution	Each solution item is documented and tracked to completion. Evidence is provided by teams owning Problem resolution items on the status and outcome of each activity.
Problem Closure and Major Problem review, if applicable.	Resolution Review and Problem Closure	A summary of all Root Cause Analysis findings and all Problem resolution activities completed is delivered. Service Management reviews the delivered summary and provides an approval or rejection. If an approval is granted, the Problem record is closed.

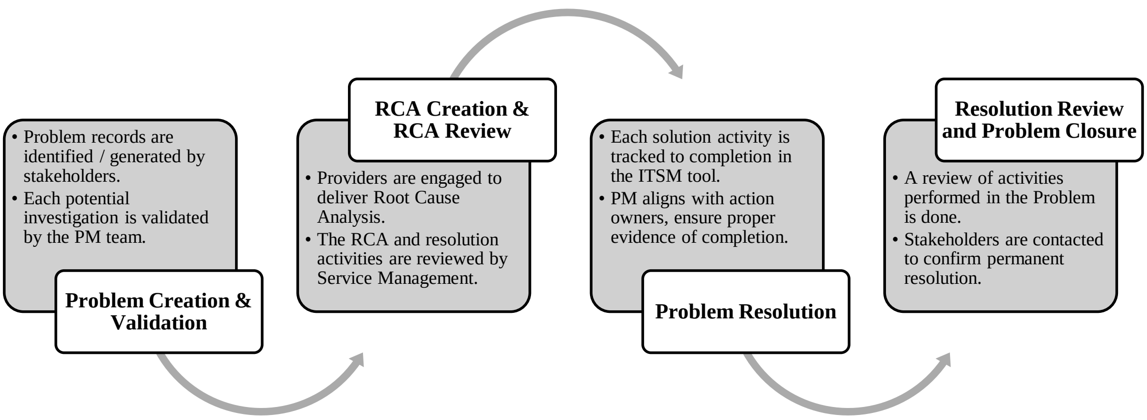

Figure 7 - Four-stage Problem lifecycle

Having validated a Problem investigation, Root Cause Analysis tasks are started. One or multiple providers collaborate to deliver an RCA document and a proposal of activities required to resolve the Problem. Although no standard RCA methodology is defined, a "5 Whys" approach is most commonly used. Here, technically experienced colleagues investigate the underlying cause of reported issues until their root cause is found; potential solutions are also listed, with respective owners and estimated due dates, aimed at permanently resolving the Problem or mitigating the risk of reoccurrence. The outcome of this effort is presented in an RCA document submitted to the organizations' Service Management community for approval or rejection, should revisions or clarification be needed.

When RCA approval is granted, each identified solution action is logged and tracked to completion in the organizations' ITSM tool. Depending on which party is responsible for implementing each activity, Providers and Service Managers deliver evidence, often from technical counterparts working in the investigation, when closing each task. Solutions can include, for example, the delivery of trainings, the updating of process documents and workflows, implementing system configuration updates through Change Management, replacing hardware or upgrading software versions.

Once all solution actions are completed, a final revision of the RCA document is done by the responsible Provider, documenting all that was found and done over the course of the investigation. This final document is submitted for approval to the Service Management community of the organization who may share it with stakeholders and customers who were impacted by the investigated issue. If an approval is granted, the Problem is considered Closed.

A team of dedicated Process Managers ensures that each activity of the four-stage Problem lifecycle is carried out according to expectations. They aid in defining action owners, create and route Problem tasks in the ITSM tool, schedule meetings for Problem validation and handling, and act in the event of escalations or overdue actions. Although they are not involved in the delivery of RCAs themselves, they ensure the process is properly driven by building connections between the business and its various Service Providers.

Based on the performed analysis, the PM process implemented in the organization is aligned with the understanding of PM presented in ITIL literature. Furthermore, observation of the four-stage Problem lifecycle being utilized indicates, to a certain extent and in select instances, that some DevOps practices are already being applied. This includes, for example, continuous Stakeholder Participation in the detection, resolution and closure phases of Problem records; reliance on a Change Management process to carry out Problem Resolution activities; utilization of automated monitoring tools to identify Problem candidates; ongoing alignment of technical resources and developers, coupled with operations teams, to investigate, diagnose and resolve Problem investigations.

Observed evidence of utilizing DevOps practices in the organizations' current PM process can be found in Table 20.

Table 20 - DevOps Practices in Org. Problem Management

		Organizational PM Practices			
		Problem Creation & Validation	RCA Creation & Review	Problem Resolution	Resolution Review and Problem Closure
DevOps Practices	Continuous Planning	The organization relies on daily PM meetings with Providers, as well as weekly forums with Service Management, to discuss the validation and progress of investigations and respective solution activities, ensuring that they are completed according to agreed timelines or adapt timelines if required.			
	Dev & Ops Feedback Loops	The Development and Operations community works closely to identify candidates for Problem investigations (often proactively, based on insight from ongoing deployments).	The Development and Operations community collaborates iteratively in the development of Root Cause analysis. Both parties are required in order to collect information from Production environments and analyze it to identify cause and propose adequate solutions.	The Development and Operations community works together to plan, implement and monitor each identified solution activity (this would include the development of bug fixes, implementation planning, and ongoing monitoring to measure success rates).	(Not observed in analysis)
	Automated Monitoring	The organization relies of monitoring tools to identify common error trends and relies on this information to initiate Proactive PM.	(Not observed in analysis)	Automated monitoring tools are utilized to measure the success of implemented resolution activities, and as an indicator of a Problem being resolved.	(Not observed in analysis)
	Stakeholder Participation	Stakeholders impacted by Incidents are encouraged to request the creation of Problem record aimed at addressing their cause.	(Not observed in analysis)	(Not observed in analysis)	Confirmation from impacted Stakeholders that a Problem has been fully resolved is actively sought after in the Resolution Review stage of the PM lifecycle.
	Change Management	(Not observed in analysis)	(Not observed in analysis)	Problem resolution activities frequently require the engagement of the Change Management process to ensures all business requirements are adhered to for implementation.	(Not observed in analysis)

5.3. Focus Group Exercise

Focus group discussions are frequently used as a qualitative approach to deepen the understanding of issues by "draw[ing] from the complex personal experiences, beliefs, perceptions and attitudes of the participants through a moderated interaction" (O.Nyumba et al., 2018). Following the completion of semi-structured interviews and a process analysis period, a focus group discussion was held with 5 select individuals from the organization, all of which were part of the group of 10 professionals previously interviewed. Referring to Table 10, these were Interviewees D, F, B, I and J. With this discussion, the aim was to establish a triangulation of data by linking insight from the focus group, the outcomes of process observation, and the results captured from semi-structured interviews.

The discussion was initiated by presenting an overview on the research questions of the case study and laying out the purpose and objectives of the exercise. A review of the findings from semi-structured interviews and process analysis, as set forth in previous chapters, was done. Afterwards, we opened the conversation for participants to comment on the presented outcomes.

On the results presented from the DevOps section of semi-structured interviews, focus group participants were generally not surprised with the observed outcomes. A comment was made by interviewee D that there was an expectation that the Test Automation practice "should have ranked higher in value" (see Table 14), as clear benefits were observed based on past experience; interviewee B commented that this was not too surprising "as test automation is currently too challenging to implement fully".

On the outcomes of the PM section only a comment was made on the topic of Known Error Management. Interviewee J provided his interpretation for this practices' low application (see Table 16) by stating that "it rates lower due to [PM] frequently lacking formal integration with the Knowledge Management process" in utilized ITSM tools.

Finally, a review of the outcomes presented on Table 17 was done, in which we had challenged interviewees to rate the significance of each DevOps practice in each stage of the ITIL PM lifecycle. Here, interviewee F questioned the high significance that was reported for Continuous Planning in the Problem Resolution stage (F4). An explanation was provided by interviewee J stating that "if we are implementing a solution as part of a

Problem investigation, it makes sense to check continuously how its deployment is being done", and that would fall in the purview of the Continuous Planning practice.

An insight presented by Interviewee B received broad consensus in the group when he noted that there seemed to be a pattern in that "the ones that are rated highest are those focused with either communication or with planning"; that is, the combinations with highest significance tend to be those in which communication or planning is done.

Having concluded the review of outcomes from semi-structured interviews, we asked focus group participants to comment on the first research question of the study: *"What DevOps practices can be applied in each stage of the PM process?"*. Here, 5 DevOps practices were highlighted. A summary of the comments on each is found in Table 21.

Table 21 - Focus Group comments on RQ1

DevOps Practice	Comment
Automated Monitoring	Participant J: "The Automated Monitoring practice can quite easily be applied to proactively gather Problem candidates from our systems (…) this is currently one of our main sources for new investigations."
Dev & Ops Feedback Loops	Participant D: "It is important for us to know the results of a new deployment and having a bridge with Operations helps us be quicker in how we respond." Participant J: "This is the most important practice that can be applied (…) the improved communication this and the Continuous Planning practice provide brings a lot of benefit and structure to the work of PM".
Stakeholder Participation	Participant B: "We need to define stakeholder, including those passively observing as well as those actively involved (…) for the detection of Problems, everyone is a valid stakeholder." Participant J: "All stakeholders can contribute to PM (…) their participation is very valuable to us".
Change Management	Participant B: "Many activities done over the course of a Problem require the involvement of Change Management to ensure they are done correctly (…) many Problems are [also] created from Changes that do not have a positive outcome."
Process Standardization	Participant I: "Organizational culture might not be leaning towards having a lot of standardization in the process, but it is needed in order for it be predictable and consistent".

Lastly, we asked participants to discuss the second research question: *"Does DevOps improve the resolution of Problems? If so, how?"*. Here, participants emphasized that professionals working in DevOps teams and those working in PM tend to have differing, yet complementary, skillsets. As one of the participants stated, "there are some activities that take place in a DevOps team that I cannot see being done in a PM team; however, DevOps team members could be part of and contribute within a PM team". A discussion on how this could be implemented practically ensued, with the concept of expanding upon the traditional PM team. Rather than being strictly composed of process managers, it could include more technically driven professionals, such as those with backgrounds in Development areas. Interviewee D highlighted that a formal reorganization of each Team might not be needed, but that it is in the "sharing of knowledge and communication between the two groups" that value is added.

Four DevOps practices were highlighted as having a role towards improving Problem resolution. A summary of the comments made for each practice is found in Table 22.

Table 22 - Focus Group comments on RQ2

DevOps Practice	Comment
Dev & Ops Feedback Loops	Participant D: "We could imagine a DevOpsProb team, where the knowledge of process is joined with technical expertise to quickly resolve Problems (…) there can still be specialization, but all working towards the same purpose." Participant B: "Communication between Problem managers and those actually developing solutions is needed to make sure things are done in an organized way and at the right time." Participant I: "Root Causes need to be found quickly and data can be lost (…) feedback is important to make sure people are on the right track to reach a conclusion."
Stakeholder Participation	Participant I: "Other processes may need to be more customer facing, but having more participation from the business in PM makes it possible to know where to focus effort, where to make priorities" Participant J: "If Stakeholders understand the process, they can be important allies over the time it takes to resolve investigations."
Change Management	Participant F: "PM benefits indirectly: If all teams are using this [Change Management], it will naturally feed into PM."
Continuous Planning	Participant J: "Some investigations are very long (…) they have complex actions that need to be consistently monitored. If planning for these Problems is not done continuously, we would lose track of what has been done, what is being done and what still needs to be completed."

In closing, participants agreed that although without formalization, some DevOps practices are already being applied in various Problem investigations, with positive results, within the organization. For example, the carrying out of detailed Root Cause Analysis was mentioned as a result of continuous feedback loops between Development and Operations; the possibility of quickening the implementation of Problem resolution activities, done through Continuous planning; or the increased identification of issues requiring PM intervention, as a result of greater Stakeholder participation. It was clear to participants that not all DevOps practices are equally relevant to a PM process, but that those which have a role in planning, communicational and collaborative efforts of teams are of significant value.

The focus group discussion was then finalized with a brief exchange on how DevOps practices could be implemented beyond specific departments. According to interviewee B there is room and reason to consider expanding the scope of these practices beyond IT sectors: “The way forward is not applying these ways of working only into the IT processes, but by using them in the organization overall.”

Chapter 6 – Conclusion

The potential role of DevOps practices in ITSM environments is, by and large, unexplored territory. The SLR carried out as part of this work, in which 36 DevOps Case Studies were reviewed, is an indication towards the need for further investigation on the subject. Only three out of the 36 case studies considered a link between DevOps practices and ITIL processes; however, none delivered specific research on DevOps' relevance for a PM process. As such, this research provided an academic contribution to the current body of knowledge by considering what DevOps practices may be applicable to a PM process, and to what extent they can lead to improved Problem resolution.

A case study was planned and carried out in an organization where both DevOps and PM are applied according to modern standards. Professionals working with various backgrounds, experiences and roles in the organization were interviewed on their understanding of DevOps, PM practices, and on how these could be used collaboratively.

Data resulting from semi-structured interviews indicates that 10 out of the 12 contemplated DevOps practices are, in terms of their application, significant in at least one stage of the Problem lifecycle. The DevOps practices of Continuous Planning, Dev & Ops Feedback Loops, Automated Monitoring, and Stakeholder Participation stand out, and were found to be highly significant to the PM process according to interviewees. The improved planning and collaboration these practices tend to enable, by allowing multiple parties to work more closely together, was frequently referred to in captured comments.

A review of the PM process implemented in the organization was performed, and practical observation done on the progress of Problem investigations. The process was found to be cohesive with current ITIL standards and practices, delivering root cause analysis and solutions to mitigate recurring Incidents and Major Incidents. According to observation, the same DevOps practices identified in semi-structured interviews as having high significance, plus Change Management, were found to already have some degree of application in the implemented PM process. For example, the Continuous Planning practice was observed in the established communication channels between the PM Team, Service Management and engaged Providers; the Automated Monitoring practice was observed in the monitoring tools implemented by the organization to proactively identify Problem candidates; the Stakeholder Participation practice was relied on during RCA and Resolution Review stages, to ensure that Problem investigations delivered appropriate solutions to known errors.

Lastly, a focus group exercise was conducted in which findings from semi-structured interviews, process analysis, and observation were reviewed. The RQs this study proposed to answer were also discussed. According to the focus group, it is clear that DevOps is relevant to the PM process; however, not all DevOps practices are equally applicable to it. On one hand, practices such as Continuous Planning, Stakeholder Participation, and Dev & Ops Feedback loops, when implemented, are found to improve the collaboration between multiple teams and departments in the organization, enabling improved Problem resolution. On the other hand, practices such as Application Prototyping, Continuous Integration and Deployment Automation are more pertinent for teams and departments working specifically in Development areas.

In summary, this study proposed to answer *what DevOps practices can be applied in each stage of the PM process*, as well as *if, and how, it can improve the resolution of Problem investigations*. Based on the activities completed over the course of the case study we conclude that 10 out of the 12 contemplated DevOps practices have varying degrees of significance in each stage of a Problem lifecycle (see Table 17); we also conclude, based on observation and the focus group activity, that at least 4 DevOps practices can directly improve the resolution of Problem investigations (see Tables 20 and 22).

It is important to note that the DevOps practices of Infrastructure as Code and Shift-Left were not widely known amongst case study participants and will be considered a limitation for this research.

In the absence of studies focused on the relationships between DevOps and ITSM, particularly PM, this research also provides a valuable contribution to professionals and practitioners alike. The insight captured on the role each DevOps practice can have for a PM process can motivate its increased application in business, generating benefits as implementation takes place.

6.1. Research Limitations

This research has some limitations. A wide variety of DevOps practices was sought to be encompassed in the case study, however some were not known nor applied by the participants that contributed in it. As such, it was not possible to conclude the potential role of said practices in the PM process. Other DevOps practices may also be proposed to have been included in the study. Furthermore, the case study relied on experience and insight from professionals working within various teams, DevOps and Process Management, within a single organization; other perspectives may exist.

6.2. Proposal for Future Work

Future work possibilities were identified over the course of this research. In it, we analyzed the scope of DevOps practices in a PM process, akin to what was done by Faustino (2018) for Incident Management. Other ITIL processes, such as Knowledge Management, Release Management and Deployment Management would also potentially benefit from being subject to this investigation. Additionally, despite providing examples of instances where DevOps practices were practically observed in a live PM process, the challenges and effort required for their implementation was not explored. Further work aimed more specifically at reviewing the implementation of DevOps practices in ITSM processes could yield additional benefits.

Bibliographical references

Abdelkebir, S., Maleh, Y., & Belaissaoui, M. (2017). An Agile Framework for ITS Management In Organizations: A Case Study Based on DevOps. *Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems*, 1–8. https://doi.org/10.1145/3167486.3167556

Aberdeen, T. (2013). Yin, R. K. (2009). Case study research: Design and methods (4th Ed.). Thousand Oaks, CA: Sage. *The Canadian Journal of Action Research*, *14*(1), 69–71. https://doi.org/10.33524/cjar.v14i1.73

Aiello, B., & Sachs, L. (2016). *Agile Application Lifecycle Management: Using DevOps to Drive Process Improvement* (1st ed.). Addison-Wesley Professional.

Albuquerque, A. B., & Cruz, V. L. (2019). Implementing DevOps in Legacy Systems. Em R. Silhavy, P. Silhavy, & Z. Prokopova (Eds.), *Intelligent Systems in Cybernetics and Automation Control Theory* (pp. 143–161). Springer International Publishing. https://doi.org/10.1007/978-3-030-00184-1_14

Alsolamy, A. A., Khan, U. A., & Khan, P. M. (2014). IT-business alignment strategy for business growth. *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, 364–366. https://doi.org/10.1109/IndiaCom.2014.6828160

Alt, R., Auth, G., & Kögler, C. (2018). Transformation of Consulting for Software-Defined Businesses – Lessons from a DevOps Case Study in a German IT Company. Em *Contributions to Management Science*. https://doi.org/10.1007/978-3-319-95999-3_19

AL-Zahran, S., & Fakieh, B. (2020). How DevOps Practices Support Digital Transformation. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*, 2780–2788. https://doi.org/10.30534/ijatcse/2020/46932020

Badinelli, R., Barile, S., Ng, I., Polese, F., Saviano, M., & Di Nauta, P. (2012). Viable service systems and decision making in service management. *Journal of Service Management*, *23*, 498–526. https://doi.org/10.1108/09564231211260396

Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective* (1st edition). Addison-Wesley Professional.

Benitez, J., Chen, Y., Teo, T. S. H., & Ajamieh, A. (2018). Evolution of the impact of e-business technology on operational competence and firm profitability: A panel data investigation. *Information & Management*, *55*(1), 120–130. https://doi.org/10.1016/j.im.2017.08.002

Bersani, M. M., Marconi, F., Tamburri, D. A., Jamshidi, P., & Nodari, A. (2016). Continuous Architecting of Stream-Based Systems. *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 146–151. https://doi.org/10.1109/WICSA.2016.26

Beulen, E. (2019). Implementing and Contracting Agile and DevOps: A Survey in the Netherlands. Em J. Kotlarsky, I. Oshri, & L. Willcocks (Eds.), *Digital Services and Platforms. Considerations for Sourcing* (pp. 124–146). Springer International Publishing. https://doi.org/10.1007/978-3-030-15850-7_7

Booch, G. (1991). *Object Oriented Design with Applications*. Benjamin/Cummings Publishing Company.

Bruza, M. (2018). *An Analysis of Multi-domain Command and Control and the Development of Software Solutions through DevOps Toolsets and Practices*. Undefined. /paper/An-Analysis-of-Multi-domain-Command-and-Control-and-Bruza/620b46ce6931895809c370769ca7a7d0fdf78271

Calder, B., Phillips, L., & Tybout, A. (1982). The Concept of External Validity. *Journal of Consumer Research*, *9*, 240–244. https://doi.org/10.1086/208920

Cannon, D., & Wheeldon, D. (2007). *ITIL Service Operation*. TSO.

Caprarelli, A., Nitto, E., & Tamburri, D. (2020). *Fallacies and Pitfalls on the Road to DevOps: A Longitudinal Industrial Study* (pp. 200–210). https://doi.org/10.1007/978-3-030-39306-9_15

Cartlidge, A., & Lillycrop, M. (2009). *An Introductory Overview of ITIL V3*. ITSMF.

Cater-Steel, A., Toleman, M., & Tan, W.-G. (2006). Transforming IT service management- The ITIL impact. *ACIS 2006 Proceedings - 17th Australasian Conference on Information Systems*.

Chen, L. (2017). Continuous Delivery: Overcoming adoption challenges. *Journal of Systems and Software*, *128*, 72–86. https://doi.org/10.1016/j.jss.2017.02.013

Colomo-Palacios, R., Fernandes, E., Soto-Acosta, P., & Larrucea, X. (2018). A case analysis of enabling continuous software deployment through knowledge management. *International Journal of Information Management*, *40*, 186–189. https://doi.org/10.1016/j.ijinfomgt.2017.11.005

Cook, T. D., & Campbell, D. T. (1979). *Quasi-Experimentation: Design and Analysis Issues for Field Settings*. Houghton Mifflin.

de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., & Brinkkemper, S. (2018). DevOps Competences and Maturity for Software Producing Organizations. Em J. Gulden, I. Reinhartz-Berger, R. Schmidt, S. Guerreiro, W. Guédria, & P. Bera (Eds.), *Enterprise, Business-Process and Information Systems Modeling* (pp. 244–259). Springer International Publishing. https://doi.org/10.1007/978-3-319-91704-7_16

de Kort, W. (2016). What Is DevOps? Em W. de Kort (Ed.), *DevOps on the Microsoft Stack* (pp. 3–8). Apress. https://doi.org/10.1007/978-1-4842-1446-6_1

De Sanctis, M., Bucchiarone, A., & Trubiani, C. (2020). A DevOps Perspective for QoS-Aware Adaptive Applications. Em J.-M. Bruel, M. Mazzara, & B. Meyer (Eds.), *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (pp. 95–111). Springer International Publishing. https://doi.org/10.1007/978-3-030-39306-9_7

Debroy, V., Miller, S., & Brimble, L. (2018). Building lean continuous integration and delivery pipelines by applying DevOps principles: A case study at Varidesk. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 851–856. https://doi.org/10.1145/3236024.3275528

Denzin, N. K., & Lincoln, Y. S. (1994). *Handbook of qualitative research* (pp. xii, 643). Sage Publications, Inc.

Di Nitto, E., Jamshidi, P., Guerriero, M., Spais, I., & Tamburri, D. (2016). *A software architecture framework for quality-aware DevOps*. 12–17. https://doi.org/10.1145/2945408.2945411

Díaz, J., Almaraz, R., Pérez, J., & Garbajosa, J. (2018). DevOps in practice: An exploratory case study. *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 1–3. https://doi.org/10.1145/3234152.3234199

Díaz, J., Pérez, J. E., Lopez-Peña, M. A., Mena, G. A., & Yagüe, A. (2019). Self-Service Cybersecurity Monitoring as Enabler for DevSecOps. *IEEE Access*, *7*, 100283–100295. https://doi.org/10.1109/ACCESS.2019.2930000

Díaz, J., Perez, J. E., Yague, A., Villegas, A., & de Antona, A. (2019). DevOps in Practice – A Preliminary Analysis of Two Multinational Companies. Em X. Franch, T. Männistö, & S. Martínez-Fernández (Eds.), *Product-Focused*

Software Process Improvement (pp. 323–330). Springer International Publishing. https://doi.org/10.1007/978-3-030-35333-9_23

Dobson, P. (1999). *Approaches to Theory Use in Interpretive Case Studies—A Critical Realist Perspective*.

Düllmann, T. F., Paule, C., & Hoorn, A. v. (2018). Exploiting DevOps Practices for Dependable and Secure Continuous Delivery Pipelines. *2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE)*, 27–30.

Duvall, P., Matyas, S., & Glover, A. (2007). *Continuous integration: Improving software quality and reducing risk* (First). Addison-Wesley Professional.

Dyck, A., Penners, R., & Lichter, H. (2015). Towards definitions for release engineering and DevOps. *Proceedings of the Third International Workshop on Release Engineering*, 3.

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

Erich, F., Amrit, C., & Daneva, M. (2014). A Mapping Study on Cooperation between Information System Development and Operations. Em A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch, & M. Raatikainen (Eds.), *Product-Focused Software Process Improvement* (pp. 277–280). Springer International Publishing. https://doi.org/10.1007/978-3-319-13835-0_21

Farroha, S., & Farroha, D. (2014). *A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment*. 288–293. https://doi.org/10.1109/MILCOM.2014.54

Faustino, J. P. C. (2018). *DevOps practices in incident management process*. https://repositorio.iscte-iul.pt/handle/10071/18294

Feitelson, D., Frachtenberg, E., & Beck, K. (2013). Development and Deployment at Facebook. *Internet Computing, IEEE*, *17*, 8–17. https://doi.org/10.1109/MIC.2013.25

Forsgren, N., & Humble, J. (2015). DevOps: Profiles in ITSM Performance and Contributing Factors. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2681906

Furfaro, A., Gallo, T., Garro, A., Saccà, D., & Tundis, A. (2016). ResDevOps: A Software Engineering Framework for Achieving Long-Lasting Complex Systems. *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 246–255. https://doi.org/10.1109/RE.2016.15

Gall, M., & Pigni, F. (2018). Leveraging DevOps for mission critical software. *AMCIS*.

Ghantous, G. B., & Gill, A. Q. (2018). DevOps Reference Architecture for Multi-cloud IOT Applications. *2018 IEEE 20th Conference on Business Informatics (CBI)*, *01*, 158–167. https://doi.org/10.1109/CBI.2018.00026

Guerriero, M., Ciavotta, M., & Gibilisco, G. P. (2015). *SPACE4Cloud: A DevOps environment for multi-cloud applications*. 29–30. https://doi.org/10.1145/2804371.2804378

Gupta, R. K., Venkatachalapathy, M., & Jeberla, F. K. (2019). Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization. *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, 30–34. https://doi.org/10.1109/ICGSE.2019.00020

Heistand, C., Thomas, J., Tzeng, N., Badger, A. R., Rodriguez, L. M., Dalton, A., Pai, J., Bodzas, A., & Thompson, D. (2019). DevOps for Spacecraft Flight Software.

2019 IEEE Aerospace Conference, 1–16. https://doi.org/10.1109/AERO.2019.8742143

Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. (2020). From Agile to DevOps: Smart Skills and Collaborations. *Information Systems Frontiers*, *22*(4), 927–945. https://doi.org/10.1007/s10796-019-09905-1

Hemon-Hildgen, A., Fitzgerald, B., Lyonnet, B., & Rowe, F. (2019). Innovative Practices for Knowledge Sharing in Large-Scale DevOps. *IEEE Software*, *PP*. https://doi.org/10.1109/MS.2019.2958900

How to Do Your Case Study. (2021, Setembro 20). SAGE Publications Ltd. https://uk.sagepub.com/en-gb/eur/how-to-do-your-case-study/book242255

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* (1st edition). Addison-Wesley Professional.

Humble, J., & Molesky, J. (2011). *Why enterprises must adopt devops to enable continuous delivery*. *24*, 6–12.

Hüttermann, M. (2012). *DevOps for Developers*. https://doi.org/10.1007/978-1-4302-4570-4

ITIL® Foundation Handbook 3rd ed. (2012). (sem data). AXELOS. Obtido 17 de Dezembro de 2020, de https://www.axelos.com/store/Book/itil-foundation-handbook-3rd-ed-2012

Jamous, N., Bosse, S., Goerling, C., Hintsch, J., Khan, A., Kramer, F., Muller, H., & Turowski, K. (2016). *Towards an IT Service Lifecycle Management (ITSLM) Concept*. 29–38. https://doi.org/10.1109/ES.2016.10

Jiménez, M., Rivera, L. F., Villegas, N. M., Tamura, G., Müller, H. A., & Gallego, P. (2019). DevOps' Shift-Left in Practice: An Industrial Case of Application. Em

J.-M. Bruel, M. Mazzara, & B. Meyer (Eds.), *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (pp. 205–220). Springer International Publishing. https://doi.org/10.1007/978-3-030-06019-0_16

Jones, S., Noppen, J., & Lettice, F. (2016). *Management challenges for DevOps adoption within UK SMEs*. 7–11. https://doi.org/10.1145/2945408.2945410

Kamuto, M. B., & Langerman, J. J. (2017). Factors inhibiting the adoption of DevOps in large organisations: South African context. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 48–51. https://doi.org/10.1109/RTEICT.2017.8256556

Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations* (Illustrated edition). IT Revolution Press.

Krishna Kaiser, A. (2018). Problem Management Adaption. Em A. Krishna Kaiser (Ed.), *Reinventing ITIL® in the Age of DevOps: Innovative Techniques to Make Processes Agile and Relevant* (pp. 201–230). Apress. https://doi.org/10.1007/978-1-4842-3976-6_8

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

Laukkanen, E., Paasivaara, M., Itkonen, J., Lassenius, C., & Arvonen, T. (2017). Towards Continuous Delivery by Reducing the Feature Freeze Period: A Case Study. *2017 IEEE/ACM 39th International Conference on Software*

Engineering: Software Engineering in Practice Track (ICSE-SEIP), 23–32. https://doi.org/10.1109/ICSE-SEIP.2017.21

Leszak, M., Perry, D., & Stoll, D. (2000). *Case study in root cause defect analysis*. 428–437. https://doi.org/10.1109/ICSE.2000.870433

Limanto, A., Khwarizma, A., Imelda, Rumagit, R., Pietono, V., Halim, Y., & Liawatimena, S. (2017). *A study of Information Technology Infrastructure Library (ITIL) framework implementation at the various business field in Indonesia*. 1–4. https://doi.org/10.1109/CITSM.2017.8089244

Luo, A., Fu, J., & Liu, J. (2016). *An impact analysis method of business processes evolution in enterprise architecture*. 733–737. https://doi.org/10.1109/PIC.2016.7949595

Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, *157*, 110384. https://doi.org/10.1016/j.jss.2019.07.083

Luz, W., Pinto, G., & Bonifacio, R. (2018). *Building a Collaborative Culture: A Grounded Theory of Well Succeeded DevOps Adoption in Practice*. https://doi.org/10.1145/3239235.3240299

Lwakatare, L. E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H. H., Bosch, J., & Oivo, M. (2016). Towards DevOps in the Embedded Systems Domain: Why is It So Hard? *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 5437–5446. https://doi.org/10.1109/HICSS.2016.671

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

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016, Agosto 21). *An Exploratory Study of DevOps: Extending the Dimensions of DevOps with Practices*.

Marrone, M., & Kolbe, L. (2011). Impact of IT Service Management Frameworks on the IT Organization. *Business & Information Systems Engineering*, *3*, 5–18. https://doi.org/10.1007/s12599-010-0141-5

Melendez, K., Dávila, A., & Pessoa, M. (2016). Information technology service management models applied to medium and small organizations: A systematic literature review. *Computer Standards & Interfaces*, *47*, 120–127. https://doi.org/10.1016/j.csi.2015.10.001

Modell, S. (2005). Triangulation between case study and survey methods in management accounting research: An assessment of validity implications. *Management Accounting Research*, *16*(2), 231–254. https://doi.org/10.1016/j.mar.2005.03.001

Mohan, V., Ben Othmane, L., & Kres, A. (2018). *BP: Security Concerns and Best Practices for Automation of Software Deployment Processes: An Industrial Case Study*. 21–28. https://doi.org/10.1109/SecDev.2018.00011

Mora, M., Marx-Gomez, J., O'Connor, R., Raisinghani, M., & Gelman, O. (2015). An Extensive Review of IT Service Design in Seven International ITSM Processes Frameworks. *International Journal of Information Technology and Systems Approach*, *8*, 69–90. https://doi.org/10.4018/ijitsa.2015010104

Nybom, K., Smeds, J., & Porres, I. (2016). On the Impact of Mixing Responsibilities Between Devs and Ops. Em H. Sharp & T. Hall (Eds.), *Agile Processes, in Software Engineering, and Extreme Programming* (pp. 131–143). Springer International Publishing. https://doi.org/10.1007/978-3-319-33515-5_11

O.Nyumba, T., Wilson, K., Derrick, C. J., & Mukherjee, N. (2018). The use of focus group discussion methodology: Insights from two decades of application in conservation. *Methods in Ecology and Evolution*, *9*(1), 20–32. https://doi.org/10.1111/2041-210X.12860

Park, H. Y., Jung, S. H., Lee, Y., & Jang, K. C. (2006). The Effect of Improving IT Standard in IT Governance. *2006 International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*, 22–22. https://doi.org/10.1109/CIMCA.2006.210

Paule, C., Düllmann, T. F., & Hoorn, A. V. (2019). Vulnerabilities in Continuous Delivery Pipelines? A Case Study. *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 102–108. https://doi.org/10.1109/ICSA-C.2019.00026

Pereira, R., & Mira da Silva, M. (2012). Designing a New Integrated IT Governance and IT Management Framework Based on Both Scientific and Practitioner Viewpoint. *International Journal of Enterprise Information Systems*. https://doi.org/10.4018/jeis.2012100101

Perera, P., Silva, R., & Perera, I. (2017). *Improve software quality through practicing DevOps*. 1–6. https://doi.org/10.1109/ICTER.2017.8257807

Perry, D., Sim, S., & Easterbrook, S. (2004). *Case studies for software engineers*. *26*, 736–738. https://doi.org/10.1109/ICSE.2004.1317512

Pietrantuono, R., Bertolino, A., Angelis, G. D., Miranda, B., & Russo, S. (2019). Towards Continuous Software Reliability Testing in DevOps. *2019 IEEE/ACM 14th International Workshop on Automation of Software Test (AST)*, 21–27. https://doi.org/10.1109/AST.2019.00009

Pilorget, L., & Schell, T. (2018). *IT Management: The art of managing IT based on a solid framework leveraging the company´s political ecosystem*. Springer Vieweg. https://doi.org/10.1007/978-3-658-19309-6

Radhakrishnan, R., Mark, K., & Powell, B. (2008). IT service management for high availability. *IBM Systems Journal*, *47*, 549–561. https://doi.org/10.1147/SJ.2008.5386521

Rajkumar, M., Pole, A. K., Adige, V. S., & Mahanta, P. (2016). DevOps culture and its impact on cloud delivery and software development. *2016 International Conference on Advances in Computing, Communication, Automation (ICACCA) (Spring)*, 1–6. https://doi.org/10.1109/ICACCA.2016.7578902

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). DevOps Adoption Benefits and Challenges in Practice: A Case Study. Em P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, & T. Mikkonen (Eds.), *Product-Focused Software Process Improvement* (pp. 590–597). Springer International Publishing. https://doi.org/10.1007/978-3-319-49094-6_44

Saarelainen, K., & Jantti, M. (2015). *Quality and human errors in IT service infrastructures—Human error based root causes of incidents and their categorization*. 207–212. https://doi.org/10.1109/INNOVATIONS.2015.7381541

Sampedro, Z., Holt, A., & Hauser, T. (2018). Continuous Integration and Delivery for HPC: Using Singularity and Jenkins. *Proceedings of the Practice and Experience on Advanced Research Computing*, 1–6. https://doi.org/10.1145/3219104.3219147

Šćekić, M., Gazivoda, M., Šćepanović, S., & Nikolić, J. (2018). Application of DevOps approach in developing business intelligence system in bank. *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, 1–4. https://doi.org/10.1109/MECO.2018.8406047

Schäfer, A., Reichenbach, M., & Fey, D. (2013). *Continuous Integration and Automation for Devops* (Vol. 170, pp. 345–358). https://doi.org/10.1007/978-94-007-4786-9_28

Schork, S., Zahid, F., Pradhan, D., Kicin, S., & Schwichtenberg, A. (2019, Maio 26). *Building an Open-Source Cross-Cloud DevOps stack for a CRM Enterprise Application: A Case Study*.

Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. *EASE*. https://doi.org/10.1145/3210459.3210465

Shahin, M., Babar, M. A., & Zhu, L. (2016). The Intersection of Continuous Deployment and Architecting Process: Practitioners' Perspectives. *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–10. https://doi.org/10.1145/2961111.2962587

Sharifi, M., Ayat, M., Rahman, A. A., & Sahibudin, S. (2008). Lessons learned in ITIL implementation failure. *2008 International Symposium on Information Technology*, *1*, 1–4. https://doi.org/10.1109/ITSIM.2008.4631627

Sharma, S., Coyne, B., Wiley, J., & Inc, S. (2015). *DevOps For Dummies, 2nd IBM Limited Edition*. 76.

Smeds, J., Nybom, K., & Porres, I. (2015). DevOps: A Definition and Perceived Adoption Impediments. Em C. Lassenius, T. Dingsøyr, & M. Paasivaara (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 166–

177). Springer International Publishing. https://doi.org/10.1007/978-3-319-18612-2_14

Šmite, D., Gonzalez-Huerta, J., & Moe, N. B. (2020). “When in Rome, Do as the Romans Do”: Cultural Barriers to Being Agile in Distributed Teams. Em V. Stray, R. Hoda, M. Paasivaara, & P. Kruchten (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 145–161). Springer International Publishing. https://doi.org/10.1007/978-3-030-49392-9_10

Soni, M. (2015). *End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery* (pp. 85–89). IEEE Computer Society. https://doi.org/10.1109/CCEM.2015.29

Ståhl, D., Mårtensson, T., & Bosch, J. (2017). The Continuity of Continuous Integration: Correlations and Consequences. *Journal of Systems and Software, 127*. https://doi.org/10.1016/j.jss.2017.02.003

Sukmandhani, A. A., Wijanarko, B. D., Gunawan, E., Pratama, D., Gaol, F. L., & Sutedja, I. (2017). Measurement effectiveness and efficiency to improve the IT services using ITSM. *2017 International Conference on Information Management and Technology (ICIMTech),* 334–339. https://doi.org/10.1109/ICIMTech.2017.8273561

Tashakkori, A., & Creswell, J. (2007). Editorial: The New Era of Mixed Methods. *Journal of Mixed Methods Research*, *1*, 3–7. https://doi.org/10.1177/2345678906293042

Tellis, W. (1997). Application of a Case Study Methodology. *The Qualitative Report*, *3*(3), 1–19. https://doi.org/10.46743/2160-3715/1997.2015

The Official Introduction to the ITIL Service Lifecycle. (2007). The Stationery Office.

UK, itSMF. (2012). *ITIL Foundation Handbook* (3rd ed.). The Stationery Office.

Varga, S., Barreto, G., & Battaglin, P. D. (2019). Increasing Information Systems Availabiliy Through Accuracy, Awareness, Completeness and Manageability of ITSM. *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–4. https://doi.org/10.23919/CISTI.2019.8760686

Virmani, M. (2015). Understanding DevOps bridging the gap from continuous integration to continuous delivery. *Fifth International Conference on the Innovative Computing Technology (INTECH 2015)*, 78–82. https://doi.org/10.1109/INTECH.2015.7173368

Wahaballa, A., Wahballa, O., Abdellatief, M., Xiong, H., & Qin, Z. (2015, Setembro 23). *Toward unified DevOps model*. https://doi.org/10.1109/ICSESS.2015.7339039

Wang, Q., Song, J., Liu, L., Luo, X., & Xinhua, E. (2010). Building IT-based incident management platform. *5th International Conference on Pervasive Computing and Applications*. https://doi.org/10.1109/ICPCA.2010.5704127

What is ITIL | IT service management. (sem data). AXELOS. Obtido 23 de Dezembro de 2019, de https://www.axelos.com/best-practice-solutions/itil/what-is-itil

Wiedemann, A., Wiesche, M., & Krcmar, H. (2019). Integrating Development and Operations in Cross-Functional Teams—Toward a DevOps Competency Model. *Proceedings of the 2019 on Computers and People Research Conference*, 14–19. https://doi.org/10.1145/3322385.3322400

Yin, R. K. (1994). Discovering the Future of the Case Study. Method in Evaluation Research. *Evaluation Practice*, *15*(3), 283–290. https://doi.org/10.1177/109821409401500309

Yun, M., Lan, Y., & Han, T. (2017). Automate incident management by decision-making model. *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*. https://doi.org/10.1109/ICBDA.2017.8078811

Zainal, Z. (2007). Case study as a research method. *Jurnal Kemanusiaan*, *9*.

Annex A – Problem Management in ITIL 2011 and ITIL4

The below key concepts and definitions were captured based on the ITIL 2011 (ITIL® Foundation Handbook, 3rd ed., 2012) and ITIL4 (ITIL® Foundation, ITIL 4 Edition, 2019) manuals and glossaries.

		ITIL 2011	ITIL4
Key Definitions	**Process Definition**	The process responsible for managing the lifecycle of all Problems.	Practice responsible for reducing the likelihood and impact of incidents by identifying actual and potential causes of incidents, managing workarounds and known errors.
	Problem	A cause of one or more incidents	A cause, or potential cause, of one or more incidents.
	Workaround	Means of reducing or eliminating the impact of an incident or problem for which a full resolution is not yet available.	A solution that reduces or eliminates the impact of an incident or problem for which a full resolution is not yet available.
	Known Error	A problem that has a documented root cause and a workaround.	A problem that has been analyzed but has not been resolved.
Process Alignment		Part of the Service Operation stage of the ITIL Service Lifecycle	PM is a service management practice which contributes heavily to the Deliver & Support and the Improve activities of the Service Value Chain.
Process Steps / Stages		The following process flow is provided in ITIL 2011 literature: • Problem detection. • Problem logging, categorization and prioritization. • Investigation and Diagnosis. • Workaround and Known Error management. • Problem resolution. • Problem closure and Major Problem review if applicable.	ITIL4 suggests the following stages as central for a PM practice: • Problem identification. • Problem control. • Error control.