Universiteit Leiden
The Netherlands

Informatica & Economie

From DevOps to DataOps:

The adoption of DevOps practices in Data Integration

Shanti Lorena Kuipers

Supervisors:
Tyron Offerman & Joost Visser

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

05/08/2024

Abstract

Introduction

The current dynamic landscape of technology and business has driven a lot of organizations to adopt agile methodologies. Organizations hoped that adopting agile methodologies would achieve shorter delivery times. However, despite faster software development on the development side, the adoption did not achieve more frequent release cycles due to isolated departmental functions. This led to the emergence of DevOps to align Development and Operations teams and ensure faster overall delivery. While DevOps is primarily focused on traditional software development teams, non-traditional development teams, such as data integration teams, might also experience the benefits of adopting DevOps practices. However, there is a lack of empirical evidence on whether DevOps practices are suitable for data integration teams. Besides the lack of empirical evidence, a company raised the question of whether DevOps practices can effectively address and fulfill specific expectations for a data integration team, leading to the following research question: How to design a guide for adopting DevOps practices in a data integration team?

Methodology

This research used a Design Science Research Methodology integrated with a case study to design and demonstrate a guide for adopting DevOps practices in a data integration team. This research interviewed eight stakeholders of a data integration team and five team members of a data integration team. The transcriptions of these interviews were analysed using a thematic approach.

Results and discussion

This research designed a guide with the use of a mapping. In this mapping, fifteen identified stakeholders' expectations for the data integration team were mapped to DevOps practices. The guide was written based on the outcomes of this mapping. To demonstrate the guide, the study interviewed the data integration team and assessed their current adopted practices to the expected practices from the guide.

The study found that most expectations of the stakeholders could be mapped to agile practices, suggesting that agile methodologies might be what stakeholders seek for a data integration team, instead of a full DevOps adoption. Additionally, this research found that despite that most of the activities of a data integration team and expectations of their stakeholders could be mapped to DevOps practices, the ones on data management could not. This shows the misalignment of DevOps and data management practices, and suggests that to fully support data integration teams, the guide should not be used on it self. Moreover, the demonstration of the guide showed that the data integration team implemented DevOps practices that are not part of the guide, suggesting that the guide should be expanded with DevOps practices based on their needs.

Conclusion

This research concludes that implementing DevOps practices can help data integration teams meet stakeholders' expectations. However, it also raises the question of whether DataOps is more suitable for a data integration team due to the data management activities.

Acknowledgements

This research would not have been possible without the support, guidance, and enthusiasm of a few people.

First, I would like to express my deepest appreciation to my first supervisor, Tyron Offerman. Your guidance and patience have been crucial for me. Without your support, I would still be perfecting my work. You also helped me overcome my fear of public speaking, a challenge I never thought I would overcome. Additionally, I would like to thank my second supervisor, Joost Visser, for his role in proofreading and evaluating my thesis.

I am also grateful to the data integration team from this study. Your enthusiasm in sharing your expertise with me was invaluable for this research, and you made great companions for much-needed coffee breaks.

Lastly, I would like to mention my partner and our cat, who provided me with emotional support and companionship during this process.

Contents

1 Introduction

The current dynamic landscape of technology and business has driven a lot of organizations to adopt agile methodologies. This adoption should lead to higher quality and shorter delivery times in the development process [6]. However, a lot of organizations have found that they were not able to achieve a more frequent release cycle, primarily because departmental functions are operating in isolation from each other [6, 14]. Agile practices are usually only adopted on the Development (Dev) side which promoted faster software development. Nevertheless, the Operations (Ops) side would also need a faster cycle to achieve a more frequent release cycle as they coordinate the actual release of software in organizations [6, 14]. Thus, even if the software has been developed fast, the business cannot get value from it [6]. Building on lean and agile practices, a new method arrived to solve the alignment issue of Development and Operation teams, called DevOps [9, 14].

While DevOps is primarily focused on traditional software development teams, non-traditional development teams, such as data integration teams, might also experience the benefits of adopting DevOps practices. Data integration teams implement techniques and strategies to enable flexible sharing and integration of data across multiple autonomous data providers, ensuring flexibility in data exchange processes [8, 23]. Their main activities revolve around data management, supported by development for their infrastructure. However, empirical evidence on whether DevOps practices are suitable for data integration teams and their specific activities could not be found.

Nowadays, data has become more important than ever, especially for data-driven organizations [39]. It is easy to understand that organizations that base their decisions on data, need it delivered fast while maintaining high quality. However, data management activities are often manual, non-optimized and error-prone. This often delays the time it takes for data to deliver value to the business [27]. DataOps was developed to solve this problem. It is aimed to shorten the end-to-end data analytic life-cycle time by introducing automation in the data collection, validation, and verification process [27]. It overlaps with DevOps components, such as automation, quality assurance, and collaboration, while it also covers data management activities.

1.1 Problem identification and motivation

This study was conducted as a case study within an e-commerce subsidiary of a global manufacturer. This company raised the question of whether DevOps practices can effectively address and fulfill specific expectations for a data integration team, providing that their data integration team currently works in a DevOps manner. DevOps practices are mainly designed for traditional software development teams, and while data integration teams develop software for their infrastructure, they are not a traditional software development team. Currently, there is a lack of empirical evidence on the adoption of DevOps practices in data integration teams. This gap raises questions about how DevOps can be effectively adopted within a data integration team or if a different approach is necessary.

Given this practical question and the lack of empirical evidence on the adoption of DevOps practices in data integration teams, this research aims to design a guide for adopting DevOps practices, which will be based on the expectations of stakeholders within the data integration team. The objective of the guide is to provide DevOps practices that address and fulfil specific

expectations for a data integration team.

This research will design and demonstrate based on the following research question and sub-questions:

`How to design a guide for adopting DevOps practices in a data integration team?`

1. What are the requirements for a data integration team?
2. Which DevOps practices are currently adopted in a data integration team?

1.2 Thesis overview

This chapter contains the introduction; Section 2 includes the background & related work; Section 3 describes the methodology used in this research; Section 4 describes the design & development of the guide; Section 5 demonstrates the designed guide; Section 6 discussed the findings of this research and provides recommendations for future work; Section 7 concludes.

2 Background & related work

This chapter outlines the background and related work to this study. subsection 2.1 will discuss a standard definition of DevOps and its central components. subsection 2.2 will continue by discussing DevOps practices and provide a table of common DevOps practices. subsection 2.3 will discuss DevOps and data integration followed by DataOps.

2.1 DevOps

While there is no formal standard definition of DevOps, multiple studies have proposed definitions [4, 9, 11, 20, 22, 42]. Blinde provided an overview of seventeen definitions found in scientific literature [4]. In their study, Jabbari et al. researched the definitions of DevOps in peer-reviewed literature and identified eight key components that characterize it. Building on these findings, they combined each identified component and proposed the following definition [20, 4]:
"DevOps is a development methodology (C4) aimed at bridging the gap (C3) between Development (Dev) and Operations (C1), emphasizing communication and collaboration (C2), continuous integration (C7), quality assurance (C8) and delivery (C5) with automated deployment (C6) utilizing a set of development practices."
Given that this proposed definition includes various components, including practices, and is rooted in peer-reviewed literature, it aligns well with the scope of this research. The components will be discussed further in the next subsection.

2.1.1 Central components of DevOps

Jabbari et al. identified eight central components of DevOps based on 44 peer reviewed studies [20]:

1. Development and Operations
2. Communication, Collaboration, Team working
3. Bridge the gap
4. Development method
5. Software delivery
6. Automated deployment
7. Continuous integration
8. Quality assurance

As mentioned in many studies, DevOps is about bridging the gap between (software) development and (IT) operations teams [5, 6, 14, 24, 25]. Research on development and operations teams revealed several issues stemming from their insufficient activity integration [18, 19, 38]. These include IT operations being excluded from requirements specification, inadequate communication and information flow between the teams, insufficient test environments, limited knowledge transfer, premature system deployment, and the lack of established operational routines before deployment [18,

19, 25, 38]. The lack of integration between these teams stems from their isolation from each other, often referred to as working in silos [6, 14]. One of the core principles of DevOps is bridging the gap between these teams by breaking down the silos and promoting cross-functional collaboration.

Communication, Collaboration and Team working

Cross-functional collaboration is part of another key component of DevOps, namely Communication, Collaboration and Team working [20]. Studies show the importance of team collaboration and efficient communication between team members for organizational success [13, 15]. Furthermore, Hermawan et al. highlight the positive effect of teamwork quality in software development [15]. Software development had already switched from traditional development methods, plan driven project management, to agile software development methods [3, 14, 40]. Agile methodologies are aimed at increasing transparency of project progress, creating usable products and services, and responding more quickly and efficiently to new or changing customer requirements [3, 40]. Agile aims to achieve this by focusing on collaboration within software development teams and with its customers [21, 40]. By extending agile methodologies to operations teams, DevOps bridges the gap between the teams and creates cross-functional teams [14].

Continuous integration/Continuous Deployment

Hemon et al. investigated the transitions between development and operation teams when transitioning from Agile to DevOps, they identified three different stages: Agile, Continuous Integration and Continuous Deployment [14]. Continuous integration (CI) is the first step from Agile to DevOps. CI focuses on integrating code changes from multiple developers into a single shared repository [41]. This approach aims to identify and correct bugs in the code early in order to reduce software release time [36, 41]. The second step is continuous deployment (CD). CD extends CI by incorporating automated testing, to ensure code quality (quality assurance), and automating the delivery of software to production as soon as all automating tests have been successfully passed (automated deployment), to achieve even faster release cycles [20, 25].

2.2 DevOps Practices

Now that the foundational components of DevOps are explored, the specific practices that organizations adopt to implement DevOps methodologies effectively will be introduced. Similar to DevOps itself, there is no standard defined or set of DevOps practices. Nonetheless many studies focused on identifying DevOps practices [4, 11, 20, 25, 34, 36]. DevOps practices are activities that are proposed to be executed in the context of DevOps [20]. Another way to interpret this is that these practices are how DevOps principles are adopted within organizations. Blinde identified 47 practices in scientific literature [4]. Table 1 provides an overview of the fourteen most common identified practices. The fourteen most common identified DevOps practices are mainly focused on automated deployment, continuous integration, quality assurance, and software delivery. However, they do not contain practices on communication, collaboration, and team working.

Table 1: Most common DevOps practices identified in Scientific Literature [4, 28]

#	Practice	Sources
1	Automated and continuous deployment throughout entire pipeline	[1, 11, 20, 24, 35, 37]
2	Make small and continuous releases	[37, 30]
3	Developers get feedback based on releases	[12, 30]
4	Create development sandboxes for minimum code deployment	[11]
5	Everything is stored as code and under version control	[11, 24, 20, 17, 12]
6	Integrated configuration management	[20]
7	Automated and continuous testing in development and staging environments	[11, 20, 17, 12, 30]
8	Reduce the time it takes to test, validate and QA code	[1]
9	Code reviews are change based	[24]
10	Automated and continuous monitoring of applications and resources	[24, 20, 2, 12]
11	Automated dashboards that include health checks and performance	[12, 20]
12	Support configurable logging that can optionally be turned on/off as needed	[12]
13	Use trunk-based development over long-lived feature branches	[24, 17]
14	Use Test driven Development where all code has unit tests	[30]

2.3 DevOps and Data Integration

While DevOps is primarily focused on bridging the gap between development and operation teams [5, 6, 14, 20], there are certain IT aspects which may not be fully encompassed within traditional development teams, such as the data integration teams [10]. Data integration teams (Data Engineers) specialize in implementing techniques and strategies to enable flexible sharing and integration of data across multiple autonomous data providers, ensuring flexibility and autonomy in data exchange processes [8, 23]. These teams have a distinct set of challenges and requirements which may not be fully addressed by all of the DevOps practices as they are customized for software development teams [4, 20, 23]. Research on the adoption of DevOps practices within data integraiton teams could not be found in scientific literature. However, research in the area of data processes and DevOps practices led to the concept DataOps [10].

2.3.1 DataOps

Few scientific studies have tried to define the concept DataOps [10, 26]. The first definition of DataOps came from Palmer who published his definition on a blog [29]. This definition included the terms communication, collaboration, integration, and automation [29], overlapping with com-

ponents of DevOps. Ereth conducted an exploratory literature review, followed by interviews with industry experts to construct a working definition of DataOps [10]. Ereth proposed the following definition [10]:
"DataOps is a set of practices, processes and technologies that combines an integrated and process-oriented perspective on data with automation and methods from agile software engineering to improve quality, speed, and collaboration and promote a culture of continuous improvement"

The process of collecting data, cleaning the data, and processing data is often manual, non-optimized and error-prone, which delays the time it takes for the data to deliver value to the business [27]. As discussed by [27], DataOps is a general process aimed to shorten the end-to-end data analytic life-cycle time by introducing automation in the data collection, validation, and verification process. [32] notes that in DataOps different teams within an organization work together to deliver value, including Data Scientists, Data Engineers, Developers and Architects, Data Governance and IT, and an Operations team. Both [27] and [32] highlight the importance of the gathering of business requirements with active involvement of all these actors.

3 Methodology

This chapter outlines the methodology used in this research. This research used a Design Science Research Methodology integrated with a case study and qualitative research methods. This research explored existing literature and used semi-structured interviews with a data integration team and a selected few of their stakeholders.

3.1 Design Science Research

As Henver and Chatterjee conceptualised [16], Design Science Research (DSR) is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts and thereby contributes new knowledge to the body of scientific evidence. The designed artifacts are, as Peffers et al. highlights, the link between IS (Information Systems) research and practice [31].

Peffers et al. proposed a set of guidelines for using Design Science Research (DSR), which are based on common process elements found in the literature [31]. These guidelines consist of six stages, which were applied in this research as follows:

1. **Problem identification and motivation**
   This initial stage is focused on defining a specific real-world problem which will justify why a solution is needed. From practice, the question was raised of whether DevOps practices can effectively address and fulfill specific expectations for a data integration team. This question and the lack of emperical evidence on adopting DevOps practices within a data integration team, motivated that a guide for adopting DevOps practices within a data integration team is needed.

2. **Objectives of a solution**
   The second stage is focused on identifying the objectives for a solution based on the real-world problem. The objective of the guide is to provide DevOps practices that address and fulfil specific expectations for a data integration team.

3. **Design and development**
   The third stage is focused on designing the artifactual solution. For this stage a set of requirements that a data integration team should fulfill was formulated; the identified requirements were mapped with DevOps practices; and a guide for adopting DevOps practices in a data integration team was designed.

4. **Demonstration**
   The fourth stage is focused on demonstrating the efficacy of the artifactual solution. To demonstrate the guide, a list of activities of a data integration team was created; the activities were mapped to DevOps practices; the implemented DevOps practices were identified; and the implemented DevOps practices were compared to the practices from the guide.

5. **Evaluation**
   The fifth stage is focused on evaluating how well the artifact supports a solution to the problem. This stage would have been an evaluation of the artifact, by a practitioner, due to

time constraints this an external evaluation was not executed. Nevertheless, this research evaluated the artifact based on the findings from the demonstration and literature.

6. **Communication**
   The final stage is where the problem and the artifact are communicated to relevant audiences. The final artifactual solution was shared with the data integration team and the stakeholders involved to ensure that they understand the proposed solution and can apply it to their current situation.

3.2 Methods

3.2.1 Interviews

During the initial stage, two rounds of interviews were conducted. One round of interviews involved semi-structured interviews with stakeholders of a data integration team which lasted approximately an hour. These interviews aimed to identify the expectations for the data integration team and to assess the current performance of the team based on these expectations. The interview consisted of open-ended questions designed to encourage participants to provide detailed descriptions of their expectations and detailed descriptions of their assessment of the current performance of the team. The questions were formulated based on the components communication, collaboration and quality of the output. In addition, the participants were asked about their experiences with data integration teams and their general expectations for data integration teams based on their experiences (Appendix A).

The other round of interviews involved semi-structured interviews with members of a data integration team which lasted approximately half an hour. These interviews aimed to identify the specific activities practiced by the team. The semi-structured approach provided the flexibility to adjust questions, provide clarifications and ask for elaborations. The interviews consisted of open-ended questions designed to encourage participants to provide detailed descriptions of their activities. These questions were formulated based on the DevOps components discussed in subsubsection 2.1.1 and focused on the specific activities performed by the team related to these components. In addition, the participants were asked about Development and Operation within their team and their experiences with DevOps (Appendix B).

3.2.2 Data analysis

This research used a thematic approach to analyse the data of both interview rounds. This involved transcribing the interviews followed by coding the transcripts and closely examining the data to identify statements and themes. For the round of interviews with stakeholders, the analysis focused on identifying expectations, current perceptions of the data integration team's performance, and other relevant remarks. The identified expectations were further categorized into general themes such as Collaboration, Communication, Data quality, and Other.

For the round of interviews with the data integration team, the focus was on identifying specific activities and notable remarks. These activities were categorized into general (DevOps) themes

such as Team working, Testing, Monitoring, Development, Stakeholder engagement, and Automation. Additionally, a separate category was created for data-specific activities that are specific to data integration.

3.3 Evaluation

The methodology of this research integrates Design Science Research with a case study. Design Science Research was used in this research to not only understand the current state and the problem, it was also used to develop an artifact aimed at improving data integration teams' performance through the application of DevOps practices. Additionally, the case study approach was crucial to gain concrete, contextual, in-depth knowledge about this specific real-world problem. It enabled this research to gain contextual insights and concrete team activities that would have been limited by the use of broader survey methods. Furthermore, the semi-structured interviews allowed structure and flexibility while providing a structured list of activities and expectations and simultaneously a more in-depth understanding of participants' perceptions. The use of this methodology enables this research to contribute both theoretically and practically. It provides the understanding of the use of DevOps Practices in data integration teams and an artifact that data integration teams can implement to meet stakeholder expectations.

4 Design & Development

This chapter outlines the design and developemnt of the guide for adopting DevOps practices in a data integration team. subsection 4.1 introduces the case study. subsection 4.2 discusses the fifteen identified expectations for a data integration (DI) team. subsection 4.3 outlines a mapping, created to align the identified expectations with DevOps practices. subsection 4.4 outlines the designed artifact, a guide for adopting DevOps practices in a data integration team.

4.1 The case

This case study examined a data integration (DI) team within an e-commerce company which is a subsidiary of a global manufacturer. This case was chosen due to their question about whether DevOps practices can effectively fulfill the specific expectations for a data integration team, providing an opportunity to explore the applicability and impact of DevOps in a non-traditional development team.

The DI team operates as part of the data department within the organization and delivers to other teams within the department, including the analytics team, the data visualization team, and the data governance team. The DI team consists of the manager of DI and five team members, consisting of four data engineers and one data architect. They work together to produce their expected outputs, and according to them, they do this using DevOps practices.

Table 2: Reported inputs from stakeholders to DI

Input to DI	**Freq**
Requirements	3
Data relationships	3
Business insights	3
Definitions	2
Meaning of data	2
Quality controls	1
Data model	1
Data structure	1
Expertise	1
Direction	1

Table 3: Reported outputs from DI to stakeholders

Output from DI	**Freq**
Data model	7
Data sets	6
Documentation	1
Technical knowledge	1
Infrastructure	1
Data warehouse	1
Data environment	1

The DI team relies on different inputs from their stakeholders to create their outputs. The stakeholders mainly provide requirements, business insights, and data relationships (Table 2). Using these inputs, the DI team creates different outputs. These outputs primarily consist of data sets and data models but can also include things like documentation and technical knowledge (Table 3).

The stakeholders rely on the outputs from the DI team to perform their day-to-day operations. For instance, the data visualization team needs the data sets to visualize it; the analytics team needs it to analyse the data; and the data governance team needs it to perform quality checks on the data (Table 4).

Table 4: Reported use of outputs by stakeholders

Use of DI output	**Freq**
Data visualisation	3
Analysis	3
Product development	2
Legal purposes	1
Communication	1
Data quality checks	1

Table 5: Reported ways of communication between stakeholders and DI

Communication with DI	**Freq**
Manager of DI	6
Unstructured communication	5
Bi-weekly meetings	4
Tickets	3

There are different ways how the stakeholders and the DI team communicate with each other about the inputs and outputs (Table 5). Most of the communication goes through the manager of DI, who communicates for the team towards the stakeholders. There are also structured bi-weekly meetings and unstructured communication between the teams. Another important method for stakeholders to communicate with the team is by creating tickets. These tickets serve as requests or issues that stakeholders raise, which the DI team then processes and addresses according to priority and feasibility.

4.2 Expectations for a data integration team

For this research, eight stakeholders were interviewed. Among them was the director of the data department, who serves as the main stakeholder for all the teams within the department, including the DI team. The remaining interviewees were members of the different stakeholder teams. These interviewees were selected based on recommendations from the manager of the DI team. Table 6 provides an overview of the roles of the interviewed stakeholders within the organization, the teams they belong to, and their years of experience working with data integration teams.

This research identified fifteen expectations for the data integration team shown in Table 7. The expectations were organised into four categories: Expectations about communication between teams; Expectations about data quality; Expectations about collaboration between teams; Other expectations. The expectations were mentioned in the transcripts in 100 instances of which 46% were related to communication between teams, 28% were related to data quality, 22% were related to collaboration between teams, and 4% were other expectations.

Table 6: Stakeholders' role and years of experience with data integration teams

ID	Role	Team	Experience with DI teams
S1	Director	Data Department	35 years
S2	Analyst	Analytics	2 years
S3	Manager	Analytics	10 years
S4	Manager	Analytics	6 years
S5	Engineer	Data Visualisation	2 years
S6	Manager	Data Visualisation	7 years
S7	Manager	Data Governance	9 years
S8	Manager	Data Governance	2 years

4.2.1 Communication between teams

Out of the expectations about communication between teams, this research found that the stakeholders expect updates on planning (23 instances), expect clear communication (16), expect updates on data related changes (5), expect communication about alerts (1), and expect updates on challenges (1). Below follows an explanation of the expectations about communication between teams:

Expect updates on planning (23). Updates on planning is the most mentioned expectation around communication between teams mentioned by participants. The stakeholders reported to expect a clear planning, updates on the changes in the planning, and updates on specific tickets that they filed. As stakeholder S5 stated: *"I think one expectation that we definitely have is let's say in terms of long term planning, when can we expect data sets or models that we are depending on for open request from the business"*

Expect clear communication (16). The participants reported to expect that the DI team communicates clearly by using a communication process, and communicating frequently, openly and proactive. Stakeholder S4 highlights the need for proactive communication: *"Of course, I also expect them to communicate with me ahead of time if they need something from me."*

Expect updates on data related changes (5). The participants, such as stakeholder S2, reported that they expect DI to give updates when something data related changes. Stakeholder S2 noted: *"When they decide on something new, like a new model, which may be affecting us or our tasks or our reports, we expect to be communicated and taken as a stakeholder by data integration and may be included in the decision-making."* The quote emphasizes the need for updates when a new data model is chosen, which may affect the workflow of the stakeholder's team.

Expect communication about alerts (1). Stakeholder S4 reported that they would ideally expect DI to communicate to them when alerts occur, as stated: *"Sometimes we are missing data for certain countries and we are flagging that which is not necessarily a big deal. Ideally they would flag*

that to us and not the other way around."

Expect updates on challenges (1). Stakeholder S7 clearly stated to expect proactive communication on challenges: *"Proactively letting us know about challenges and timelines is one expectation."*

4.2.2 Data quality

Out of the expectations about data quality, this research found that the stakeholders expect high quality data (23), expect testing on data products before releasing to production (3), and expect maintenance of data products (2). Below follows an explanation of the expectations about data quality:

Expect high quality data (23). High quality data is the most mentioned expectations around data quality mentioned by participants. They reported that they expect accuracy and reliability in the data, but also that the data is expected to be up to date and needs to match the source data. As stakeholder S1 stated: *"It [data] needs to be in time it needs to be accurate, it needs to fulfill our seven C's for our data, principles that we have established inside of the team so that we can meet them and ensure that these are delivered."*

Expect testing on data products before releasing to production (3). The participants reported that it is important for them that the data products are tested before they are released to production. Automated validation and testing on data model were mentioned by the participants as expectations regarding this.

Expect maintenance of data products (2). An interest was shown into the maintenance of the data products by participants, including efficient maintenance, and incorporating monitoring and alerts. As stakeholder S7 stated: *"The environment is also making it complicated but, from my own experience, the challenge is to find the middle ground between maintenance. Because maintenance eats up capacities and maintenance work can also be very much demotivating for a team if you need to do it too much"*
This quote emphasizes the balance that must be met between performing necessary maintenance tasks and avoiding the excessive allocation of resources to these tasks, which can be demotivating for the team. Efficient maintenance involves optimizing these tasks to ensure they are manageable and do not hinder the team's productivity.

4.2.3 Collaboration between teams

Out of the expectations about collaboration between teams, this research found that the stakeholders expect structured collaboration on data products (10), expect structured collaboration between teams (5), expect understanding of the business context around data products (5), and expect documentation (2). Below follows an explanation of the expectations about collaboration between teams:

Expect structured collaboration on data products (10). A structured collaboration on data products was the most mentioned expectation around collaboration between teams mentioned by participants. They reported to expect collaboration in creating a daily data refresh and a reporting table as well

as collaboration on a data model/set, input on data product improvements and that the output meets goals the goals and strategies of the department. Such as stakeholder S2 clearly stated: *"Maybe some suggestions to us about how we can use it [query] in a more efficient way and maybe performance issue exist in our own query. Maybe they can also suggests some improvements about it. So, we need some collaboration around this."*

Expect structured collaboration between teams (5). The stakeholders mentioned that they expect a structured collaboration between their teams and the DI team. A structured collaboration approach, clear definitions of roles and responsibilities of team members, and collaboration in looking for solution were reported by them as expectations regarding this. Stakeholder S8 highlights this: *"I think at the end of the day, maybe there should be a more structure. As I said, the collaboration and more synergy between the different teams."*

Expect understanding of the business context around data products (5). The DI team is expected to have an understanding of the business context around data products according to the participants. They reported to expect that the DI team understands the business, understands use cases, and proactively asks questions about requirements. Stakeholder S3 stated that DI needs to understand the use cases of the data: *"They also need to communicate with the outside and understand the use cases that the data is being used for."*

Expect documentation (2). As reported by the participants, the stakeholders expect documentation from DI. They mentioned that they expect to have clear documentation on the data platform for users and to have documentation on definitions. As stakeholder S2 stated the need for documentation for stakeholders (users): *"We need clear documentation, but the audience shouldn't be data integration, the audience should be stakeholders accordingly. We need the documentation too because we need to encourage people to use data platform and in current way it's very hard. It's just a few people using the data platform because it's not easy to understand."*

4.2.4 Other expectations

Out of the other expectations, this research found that the stakeholders expect knowledge sharing (2), expect DI to be aware of data privacy and compliance (1), and expect to influence cultural change (1). Below follows an explanation of the other expectations:

Expect knowledge sharing (2). Two instances were reported in the interviews where stakeholders highlighted that they expect the DI team to share knowledge. One of these instances, was by stakeholder S1, noting: *"What they now need to do in order to enter into the next phase is to really work more independent, and share knowledge."*

Expect DI to be aware of data privacy and compliance (1). Stakeholder S7 highlighted the importance of data privacy and compliance, where the participant expects the DI team to be aware of: *"And [expect them] proactively being aware of data privacy and other compliance things and sharing them with us if they see something which is the case."*

Expect to influence cultural change (1). Stakeholder S1 highlighted that DI can and is expected to

influence cultural change by the delivery of high quality data: *"If you want to do a cultural change as well that people start to trust data. Even if the data says something that is contradicting their gut feeling, as long as data says the same that your gut feeling does, people are easy to follow data. But the moment data tells you your gut feeling is wrong and you should do it differently. There is where the cultural change comes into place, and this is where data integration from my standpoint, plays a crucial role in introducing that by delivering a whole high quality of data."*

Table 7: Overview of expectations reported by stakeholders

Category	Expectation	Sub-expectations	Freq
Communication between teams	Expect updates on planning	Expect updates on planning	11
		Expect updates on changes in planning	4
		Expect a clear planning	4
		Expect updates on specific tickets	4
	Expect clear communication	Expect a clear communication process	6
		Expect frequent communication	4
		Expect proactive communication	4
		Expect open communication	2
	Expect updates on data related changes		5
	Expect communication about alerts		1
	Expect updates on challenges		1
Data quality	Expect high quality data	Expect data to be accurate	6
		Expect data to be reliable	5
		Expect data to match the source data	5
		Expect high quality data	4
		Expect data to be up to date	3
	Expect testing on data products before releasing to production	Expect automated validation	1
		Expect testing before releasing to production environment	1
		Expect testing on data model	1
	Expect maintenance of data products	Expect efficient maintenance of data products	1
		Expect monitoring and alerts	1
Collaboration between teams	Expect structured collaboration on data products	Expect input on data product improvements	3
		Expect output to meet goals and strategies of department	3
		Expect to collaborate on data model/set	2
		Expect collaboration in creating daily data refresh	1
		Expect collaboration in creating reporting table	1
	Expect structured collaboration between teams	Expect to collaboratively look for solutions	2
		Expect collaboration between teams	1
		Expect structured collaboration approach	1
		Expect clear definitions of roles and responsibilities of team members	1
	Expect understanding of business context around data products	Expect to understand the business	2
		Expect to proactively ask questions about requirements	2
		Expect to understand use cases	1
	Expect documentation	Expect to have clear documentation of data platform for users	1
		Expect to have documentation on definitions	1
Other	Expect knowledge sharing		2
	Expect them to be aware of data privacy and compliance		1
	Expect to influence cultural change		1

4.3 Mapping stakeholder expectations to DevOps practices

This section provides an overview of the mapping between expectations and DevOps practices. Additionally, the expectations that could not be mapped are discussed.

Based on the fifteen identified stakeholder expectations (subsection 4.2) and the fourteen most common DevOps practices from the research by Blinde [4, 28] (subsection 2.2), a mapping was created between expectations for a DI team and DevOps practices. As only few expectations could be mapped with the most common DevOps practices, two less common practices from the research of Blinde [4] are introduced which are focused on collaboration.

- ***Practice 15.*** Use Agile and LEAN practices (e.g. sprint planning and requirements engineering)
- ***Practice 16.*** Collaborate with stakeholders often and provide overall visibility into the project

Table 8 displays the fifteen identified expectations on the y-axis and the DevOps practices on the x-axis. Mappings between them are indicated with a cross.

4.3.1 Expected DevOps practices

Table 8 shows that out of the fifteen identified expectations, twelve could be mapped with the DevOps practices. It also shows that out of the sixteen DevOps practices, nine applied to the expectations. Below follows an elaboration of the DevOps practices and their mapped expectations:

Practice 2: Make small and continuous releases.
This practice involves making small and continuous releases. The expectation mapped to this practice is 'expect maintenance of data products'. Implementing this practice, ensures that the application is kept up-to-date.

Practice 3: Developers get feedback based on releases.
This practice involves developers receiving feedback based on releases. Expectations mapped to this practice include 'expect structured collaboration on data products' and 'expect understanding of business context around data products'. This practice allows stakeholders to provide feedback on each release, fostering ongoing collaboration on data products. Furthermore, receiving feedback from stakeholders helps developers to better align their work to meet the goals and strategies of the department.

Practice 6: Integrated configuration management.
This practice involves integrated configuration management. The expectation mapped to this practice is 'expect documentation'. By implementing this practice, the team can ensure the automatic creation of team documentation with every change in the code base.

Practice 7: Automated and continuous testing in development and staging environments.
This practice ensures that data products are consistently validated in both development and staging environments. The expectations mapped to this practice is 'expect testing on data products before

Table 8: A mapping of stakeholder expectations to DevOps practices from [4, 28].

	1 Automated and continuous deployment throughout entire pipeline	2 Make small and continuous releases	3 Developers get feedback based on releases	4 Create development sandboxes for minimum code deployment	5 Everything is stored as code and under version control	6 Integrated configuration management	7 Automated and continuous testing in development and staging environment	8 Reduce the time it takes to test, validate, and QA code	9 Code reviews are changed based	10 Automated and continuous monitoring of applications and resources	11 Automated dashboards that include health checks and performance	12 Support configurable logging that can optionally be turned on/off as needed	13 Use trunk-based development over long-lived feature branches	14 Use Test driven Development where all code has unit tests	15 Use Agile and LEAN practices (e.g. sprint planning and requirements engineering)	16 Collaborate with stakeholders often and provide overall visibility into the project
Expect updates on planning															x	x
Expect clear communication															x	x
Expect updates on data related changes															x	x
Expect communication about alerts										x					x	x
Expect updates on challenges															x	x
Expect high quality data																
Expect testing on data products before releasing to production							x							x		
Expect maintenance of data products		x								x	x					
Expect structured collaboration on data products			x													x
Expect structured collaboration between teams															x	x
Expect understanding of business context around data products			x												x	x
Expect documentation						x									x	
Expect knowledge sharing															x	x
Expect DI to be aware of data privacy and compliance																
Expect to influence cultural change																

releasing to production'.

Practice 10: Automated and continuous monitoring of applications and resources.
This practice involves automated and continuous monitoring of applications and resources. Expectation mapped to this practice include 'expect maintenance of data products' and 'expect communication about alerts'. For maintenance of a data product, it is important to monitor the application and resources involved. Additionally, monitoring is needed to receive automated alerts when there are issues with the applications and resources involved.

Practice 11: Automated dashboards that include health checks and performance.
This practice involves automated dashboards that include health checks and performance. The

expectation mapped to this practice is 'expect maintenance of data products'. Dashboards provide valuable insights into the performance and health of the data products.

Practice 14: Use Test driven Development where all code has unit tests.
This practice involves using Test Driven Development (TDD) where all code has unit tests. The expectation mapped to this practice is 'expect testing on data products before releasing to production'. This practice ensures that each component of the data product is thoroughly tested before release to production.

Practice 15: Use Agile and LEAN practices (e.g. sprint planning and requirements engineering).
This practice involves the use of Agile and LEAN practices. Expectations mapped to this practice include 'expect updates on planning', 'expect clear communication', 'expect updates on data related changes', 'expect communication about alerts', 'expect updates on challenges', 'expect structured collaboration between teams', 'expect understanding of business context around data products', 'expect documentation', and 'expect knowledge sharing'.

Practice 16: Collaborate with stakeholders often and provide overall visibility into the project.
This practice involves collaborating often with stakeholders and providing overall visibility into the project. Expectations mapped to this practice include 'expect updates on planning', 'expect clear communication', 'expect updates on data related changes', 'expect communication about alerts', 'expect updates on challenges', 'expect structured collaboration on data products', 'expect structured collaboration between teams', 'expect understanding of business context around data products', and 'expect knowledge sharing'.

4.3.2 Expectations beyond DevOps practices

Following the mapping, out of the fifteen expectations, three have not been mapped to any of the DevOps practices.

- *Expect high data quality.* This expectation was not directly mapped with any of the DevOps practices. Ensuring high data quality involves data checks and validations, which are not directly covered by the DevOps practices.
- *Expect DI to be aware of data privacy and compliance.* This expectation was not directly mapped with any of the DevOps practices. The DevOps practices do not explicitly cover regulatory requirements and policy adherence related to data.
- *Expect to influence cultural change.* This expectation was not directly mapped with any of the DevOps practices. Cultural change involves processes that go beyond the specific technical and procedural practices covered by the DevOps practices.

4.4 Guide for adopting DevOps practices in a data integration team

This guide aims to aid data integration teams in adopting DevOps practices. While DevOps is primarily focused on traditional software development teams, non-traditional development teams, such as data integration teams can also experience the benefits of adopting DevOps practices. By implementing the DevOps practices discussed in this guide, a data integration team can meet the most common stakeholder expectations and thus experience benefits from adopting DevOps practices.

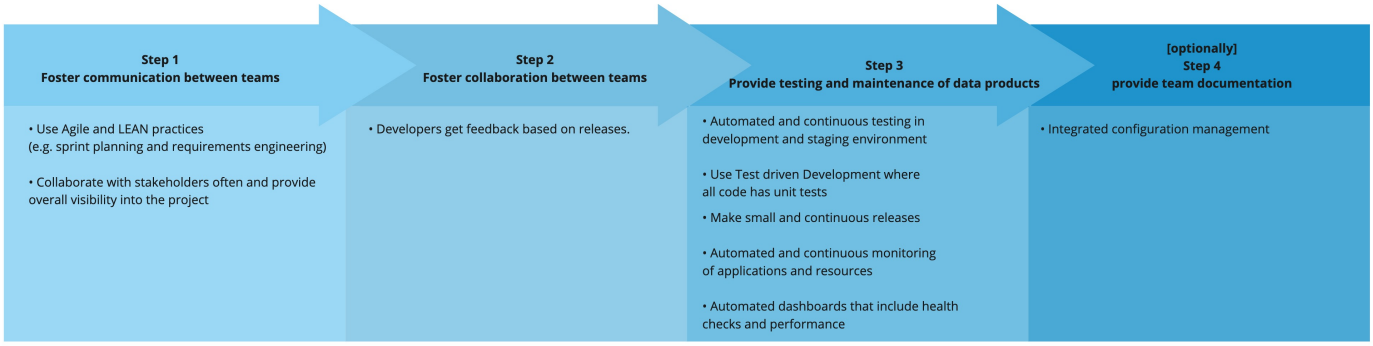

Figure 1: Visualization of the steps in the guide and the proposed DevOps practices

Step 1: Foster communication between teams

The most significant expectation from stakeholders was the need for communication between teams. This includes having clear communication and providing the stakeholders with updates. For a data integration team to meet the expectations, they should first implement the following two DevOps practices:

- Use Agile and LEAN practices (e.g. sprint planning and requirements engineering)
- Collaborate with stakeholders often and provide overall visibility into the project

These two practices promote and enhance communication between teams.

Step 2: Foster collaboration between teams

The practices from step 1 provide the data integration team with the ability to meet the expectations of communication between team. Additionally, they encourage collaboration between teams, which is another significant expectation from stakeholders. This includes the need for structured collaboration on data products and the need for understanding of the business context around data products. For a data integration team to further meet these expectations, they should follow the first set of DevOps practices by the following:

- Developers get feedback based on releases.

This practice ensures that data integration teams get feedback from stakeholders on releases. This encourages collaboration on data products and for the data integration team to receive more business context on data products.

Step 3: Provide testing and maintenance of data products

Another important aspect for stakeholders, is the quality of the data products. They expect that the data integration team, tests the products before releasing them to production, and that they are maintained. The following two practices should be implemented to ensure testing on the data products before releasing to production:

- Automated and continuous testing in development and staging environment
- Use Test driven Development where all code has unit tests

The following three practices should then be implemented to ensure maintenance of the data products:

- Make small and continuous releases
- Automated and continuous monitoring of applications and resources
- Automated dashboards that include health checks and performance

[optionally] Step 4: provide team documentation

The last expectation that is not yet covered by the previous steps is expect documentation, specifically team documentation. By implementing the following practice, the team can automatically ensure team documentation with every change in the code base:

- Integrated configuration management

Assess the Applicability for Your Data Integration Team

To determine how applicable this guide is for your data integration team, a few steps need to be taken beforehand.

1. List all activities of the data integration team.
   One can list all activities of the team either by observing the team, by interviewing the team, or by doing workshops with the team.

2. Determine the current adopted DevOps practices.
   The next step is to determine which DevOps practices the team currently implemented. This can be achieved by mapping the team's activities to DevOps practices.

3. Collect feedback from stakeholders.
   By collecting feedback from stakeholders on the current performance of the team, gaps can be identified on the categories of the expectations covered in this guide.

4. Determine the gaps.
   Based on the feedback, determine which expectations the team still needs to meet. Assess whether the team implemented the DevOps practices of this expectation. If they still need to implement practices, this guide is applicable.

By following these steps, you can measure how well this guide fits your data integration team's needs and identify areas for improvement.

5 Demonstration

This chapter outlines the demonstration of the guide. subsection 5.1 shows the 48 identified activities currently used by the DI team. subsection 5.2 discusses the DevOps practices that are currently implemented by the team. subsection 5.3 addresses the gaps between the DI team's current implemented practices and expected practices from the guide.

5.1 Activities of a data integration team

To demonstrate the designed guide, five team members of the data integration team were interviewed to investigate which DevOps practices they implemented and whether their activities include practices that are not part of DevOps but specific to data integration (Table 9). Due to circumstances the manager of the team could not be interviewed.

Table 9: Role of the team members and their years of experience with DevOps

ID	Role	Experience with DevOps
T1	Data engineer	5 years
T2	Data engineer	5 years
T3	Data engineer	10 years
T4	Data architect	8 years
T5	Data engineer	5 years

This research identified 48 activities that the DI team reported to incorporate in their current way of working, an overview is shown in Table 10. Out of the 48 activities, eleven were categorized as data specific activities. The other 37 were categorized into Team working (9 activities), Testing (7), Monitoring (7), Development (6), Stakeholder Engagement (5), and Automation (3).

- *Data specific (11).* The data specific activities entail reported activities that are used around data. This includes data checks and validations, updating and deploying seed files, automated data refreshing and ingestion, data processing pipeline and data modeling standardization.
- *Team working (9).* The team working activities entail activities which allow the DI team to work together. This includes reported team meetings such as sprint reviews, daily check, retrospective meetings, and demo sessions; reported knowledge sharing activities such as user and team documentation and architecture and design discussions; 2 week sprints; and pair programming.
- *Testing (7).* The testing activities entail reported activities around testing of the data products. This includes unit testing, automated tests on code and validation, system integration testing, development environment testing, stakeholder validation, and user acceptance testing.

- *Monitoring (7).* The monitoring activities entail reported activities around monitoring of the data products. This includes event monitoring activities such as cluster logging, alerts, and notifications on issues. Additionally, it includes performance monitoring activities such as the use of workflow dashboards and progress monitoring.
- *Development (6).* The development activities entail reported activities around the development process of the data products. This includes activities such as the use of repositories and version control, code reviews, pull requests, peer reviews on pull requests, and user stories.
- *Stakeholder Engagement (5).* The stakeholder engagement activities entail reported activities that the DI teams does together with the stakeholder. This includes regular meetings such as biweekly alignment sessions, weekly meeting, and weekly Scrum of Scrums. Additionally, these activities encompass workshops and ticket registration.
- *Automation (3).* Automation refers to reported activities that do not require manual control and rely on the use of pipelines.

Table 10: A list of activities reported by the interviewed DI team.

Category	Activity	Freq
Data specific	Data quality checks	3
	Automated data refreshing	2
	Post-transformation data validation	1
	Basic Null and integrity checks	1
	Consistency checks	1
	Source validation	1
	Relational and primary key validation	1
	Automated data ingestion	1
	Data modeling standardization	1
	Data processing pipeline	1
	Updating and Deploying Seed Files	1
Team working	Demo sessions	4
	User documentation	4
	Team documentation	4
	2 week sprints	3
	Sprint reviews	2
	Daily check	1
	Retrospective meetings	1
	Pair programming	1
	Architecture and design discussions	1
Testing	Development environment testing	3
	Stakeholder validation	3
	Unit testing	2
	Automated tests on code	1
	Automated validation	1
	System integration testing	1
	User acceptance testing	1

Category	Activity	Freq
Monitoring	Automated Notification on Issues	2
	Failed test alerts	1
	Automated alerts	1
	Failing pipeline alerts	1
	Cluster logging	1
	Workflow dashboards	1
	Progress monitoring	1
Development	Version control	4
	Pull requests	3
	Peer reviews on pull requests	3
	User stories	2
	Code review	1
	Repositories	1
Stakeholder Engagement	Biweekly alignment sessions	2
	Weekly meeting	1
	Weekly Scrum of Scrums	1
	Workshops	1
	Ticket registration	1
Automation	Testing pipeline	2
	Development pipeline	1
	Automated deployment pipeline	1

5.2 Mapping DI activities to DevOps practices

This section outlines which DevOps practices are implemented by the data integration team based on the identified activities. Additionally, the activities that could not be mapped with DevOps practices are discussed.

5.2.1 DevOps practices of the DI team

Out of the sixteen DevOps practices, thirteen were mapped with the activities of the DI team, see section B for the mapping of activities to DevOps practices. This subsection will discuss which activities are mapped to the DevOps practices and assess whether the team has fully implemented these practices. Table 11 shows an overview of the mapped practices and their implementation by the team. Below follows an elaboration of the DevOps practices and their mapped activities:

Table 11: Overview of the implementation of DevOps practices from [4, 28].

#	**Practice**	**Implemented**
1	Automated and continuous deployment throughout entire pipeline	Partially implemented
2	Make small and continuous releases	Not defined
3	Developers get feedback based on releases	Implemented
4	Create development sandboxes for minimum code deployment	Implemented
5	Everything is stored as code and under version control	Implemented
6	Integrated configuration management	Partially implemented
7	Automated and continuous testing in development and staging environments	Implemented
8	Reduce the time it takes to test, validate and QA code	Implemented
9	Code reviews are change based	Implemented
10	Automated and continuous monitoring of applications and resources	Implemented
11	Automated dashboards that include health checks and performance	Implemented
12	Support configurable logging that can optionally be turned on/off as needed	Not defined
13	Use trunk-based development over long-lived feature branches	Not defined
14	Use Test driven Development where all code has unit tests	Implemented
15	Use Agile and LEAN practices (e.g. sprint planning and requirements engineering)	Implemented
16	Collaborate with stakeholders often and provide overall visibility into the project	Partially implemented

Practice 1: Automated and continuous deployment throughout entire pipeline.
This practice involves automated and continuous deployment throughout the entire pipeline. The activities mapped to this practice are 'updating and deploying seed files,' which is partly manual since the updating part remains manual. While deploying seed files is integrated into the deployment pipeline, updating these seed files remains a manual process. As team member T3 stated, *'Our deployment automation isn't complete. Things like updating seed files for configuration and deploying, that's also manual work.'*

Additionally, 'Automated deployment pipeline,' which is still not fully automated as deployments must be triggered manually. Team member T3 stated, *'The current deployments are triggered manually, so there are no continuous deployments.'* Aside from updating seed files, four out of five DI team members indicated that their deployment pipeline is not fully automated, as it still requires a manual trigger. Therefore, while elements of the practice are in place, the lack of full automation indicates that the practice is not fully implemented.

Practice 3: Developers get feedback based on releases.
This practice involves developers receiving feedback based on releases. The activities mapped to this practice include 'stakeholder validation,' 'user acceptance testing,' and 'demo sessions.' These activities provide structured opportunities for stakeholders and users (the stakeholders in this case) to review and provide feedback on the software. As these activities ensure feedback on releases, it is assumed that this practice is implemented.

Practice 4: Create development sandboxes for minimum code deployment.
This practice involves creating development sandboxes for minimum code deployment. The activity mapped to this practice is 'development environment testing.' By setting up development sandboxes, the team can test code changes in isolated environments before integrating them into the main codebase. This allows for thorough testing and validation. If the development environment testing is conducted in these sandboxes, it indicates that the practice is implemented. However, based on the interviews, it is not sure if the development environments are isolated sandboxes. Nevertheless, this research assumes that the team did implement this practice.

Practice 5: Everything is stored as code and under version control.
This practice involves storing everything as code and under version control. The activities mapped to this practice are 'repositories' and 'version control.' Repositories are used to store code and configuration files, ensuring that all changes are tracked and managed systematically. Version control systems, on the other hand, maintain a history of changes, allowing teams to manage different versions of code and collaborate effectively. As both activities are applied by the team, it is assumed that this practice is implemented.

Practice 6: Integrated configuration management.
This practice involves integrated configuration management. The activities mapped to this practice include 'team documentation,' 'cluster logging,' 'repositories,' 'version control,' and 'pull requests.' Team documentation ensures that configuration changes are documented within the team. Cluster logging provides insights into the performance and configuration of clusters, which is needed for managing configurations. Repositories and version control help in managing and tracking configuration files. Pull requests are used to review and integrate configuration changes. As this research does not contain a formal definition for this practice, it is not clear whether this practice is fully implemented.

Practice 7: Automated and continuous testing in development and staging environments.
This practice involves automated and continuous testing in development and staging environments. The activities mapped to this practice include 'unit testing,' 'automated tests on code,' 'automated validation,' and 'system integration testing.' Unit testing focuses on verifying individual components of the code to ensure they work correctly. Automated tests on code run tests automatically as code

changes. Automated validation ensures that the code meets predefined requirements and standards continuously. System integration testing verifies that different systems work together. As all of the applied activities include the main component of this practice, this research assumes that this practice is implemented.

Practice 8: Reduce the time it takes to test, validate and QA code.
This practice involves reducing the time it takes to test, validate, and quality assure code. The activities mapped to this practice are 'development pipelines' and 'testing pipelines.' Development pipelines automate the process of integrating and testing code changes, ensuring that these processes are performed quickly. Testing pipelines focus on running automated tests, ensuring that code quality is validated in quickly. By applying these pipelines, the time required for testing, validation, and quality assurance is minimized. As the team applied these activities, it is assumed that this practice is implemented.

Practice 9: Code reviews are change based.
This practice involves having change-based code reviews. The activities mapped to this practice include 'code review' and 'peer review on pull requests.' Code reviews involve reviewing code changes. Peer reviews on pull requests also involve reviewing and providing feedback on proposed changes before they are merged into the main codebase. Both of these activities are based on change-based code reviews, thus it is assumed that this practice is implemented.

Practice 10: Automated and continuous monitoring of applications and resources.
This practice involves automated and continuous monitoring of applications and resources. The activities mapped to this practice include 'cluster logging,' 'failed test alerts,' 'automated alerts,' 'failing pipeline alerts,' and 'automated notifications on issues.' Cluster logging continuously monitors the performance of application clusters. Failed test alerts notify the team of issues detected during automated testing. Automated alerts and failing pipeline alerts notify the team members of any critical issues or failures in the deployment process. Automated notifications on issues ensure that team members are notified automatically when issues occur. These activities ensure automated and continuous monitoring of applications and resources, thus it is assumed this practice is implemented.

Practice 11: Automated dashboards that include health checks and performance.
This practice involves automated dashboards that include health checks and performance. The activities mapped to this practice are 'workflow dashboards' and 'progress monitoring.' Workflow dashboards provide a visual representation of how workflows are performing including health checks. Progress monitoring involves monitoring tasks, giving insights into performance and potential issues. It is assumed that by this practice is implemented by the use of these activities.

Practice 14: Use Test driven Development where all code has unit tests.
This practice involves using Test Driven Development (TDD) where all code has unit tests. The activity mapped to this practice is 'unit testing.' However, based on the interviews, it is not clear whether unit testing is performed on all code. Nevertheless, this research assumes that this practice is implemented.

Practice 15: Use Agile and LEAN practices (e.g. sprint planning and requirements engineer-

ing).
This practice involves the use of Agile and LEAN practices. The activities mapped to this practice include 'sprint reviews,' 'daily check,' 'retrospective meetings,' 'demo sessions,' 'pair programming,' '2-week sprints,' 'architecture and design discussions,' 'user documentation,' 'team documentation,' 'biweekly alignment sessions,' 'weekly scrum of scrums,' 'workshops,' and 'ticket registration.' Scrum, an Agile framework, includes 2 week sprints, where each sprint contains regular meetings. These regular meetings are daily (daily check), weekly (weekly scrum of scrums), and biweekly (sprint reviews or demo sessions, retrospective meetings, biweekly alignment session). Furthermore, pair programming, documentation, workshops, and ticket registration are in line with Agile and LEAN principles. As both Agile and LEAN practices are applied, it is assumed that this practice is implemented.

Practice 16: Collaborate with stakeholders often and provide overall visibility into the project.
This practice involves collaborating often with stakeholders and providing overall visibility into the project. The activities mapped to this practice include 'sprint reviews,' 'demo sessions,' 'biweekly alignment sessions,' 'weekly meetings,' and 'weekly scrum of scrums.' Sprint reviews are held at the end of each sprint, where the team can demonstrate their product to the stakeholders which is essentially a demo session. Biweekly alignment sessions are sessions with different teams, or departments to align on tasks. Weekly meetings provide frequent communication with the stakeholders. Weekly scrum of scrums are for large projects which involve multiple scrum teams to ensure that all teams are aware of the progress. However, based on the interviews it did not become clear whether these activities include all different stakeholder teams, if they occur often, and if overall visibility into the project is provided. Based on this, this research can not assume that this practice is fully implemented.

5.2.2 Activities beyond DevOps practices

Ten out of the 48 activities could not be mapped with any of the DevOps practices. These unmapped activities are primarily data-specific. Since DevOps practices are mainly focused on software development and deployment, they do not include activities related specifically to data, which lies outside the scope of software development processes.

- *Data checks and validations.* The activities involving data checks and validations could not be mapped to any of the DevOps practices. The DevOps practices do include testing, however as they are focused on software testing they do not explicitly cover data checks and validations.
- *Automated data refreshing and ingestion.* Automated data refreshing and automated data ingestion could not be mapped to any of the DevOps practices. These activities are focused on ensuring that the data is up to date and collected, which does not align with the DevOps practices.
- *Data modeling standardization.* This is a specific data activity which includes creating consistent, uniform data models. The DevOps pracices do not include specific data modeling activities.
- *Data processing pipeline.* A data processing pipeline involves automated processing and transforming of data. It does not directly map to the DevOps practices as there are no specific

practices involved in processing and transforming data.

5.3 Gaps between team practices and expected practices

Table 12: Overview of the implementation of expected DevOps practices from [4, 28].

#	**Practice**	**Implemented**	**Expected**
1	Automated and continuous deployment throughout entire pipeline	Partially	No
2	Make small and continuous releases	Not defined	Yes
3	Developers get feedback based on releases	Yes	Yes
4	Create development sandboxes for minimum code deployment	Yes	No
5	Everything is stored as code and under version control	Yes	No
6	Integrated configuration management	Partially	Yes
7	Automated and continuous testing in development and staging environments	Yes	Yes
8	Reduce the time it takes to test, validate and QA code	Yes	No
9	Code reviews are change based	Yes	No
10	Automated and continuous monitoring of applications and resources	Yes	Yes
11	Automated dashboards that include health checks and performance	Yes	Yes
12	Support configurable logging that can optionally be turned on/off as needed	Not defined	No
13	Use trunk-based development over long-lived feature branches	Not defined	No
14	Use Test driven Development where all code has unit tests	Yes	Yes
15	Use Agile and LEAN practices (e.g. sprint planning and requirements engineering)	Yes	Yes
16	Collaborate with stakeholders often and provide overall visibility into the project	Partially	Yes

This section discusses the gaps between the expected DevOps practices (subsection 4.3), and the currently implemented DevOps practices (subsection 5.2).

5.3.1 Demonstrating the guide

Table 12 shows an overview of the DevOps practices, whether they are implemented by the team, and whether they are expected based on the expectations in the guide. Out of nine expected DevOps practices covered in the guide, the team implemented six fully, two partially, and of one it is not defined whether they have implemented it. Below follows a discussion and solution of the gaps by walking through the designed guide:

Step 1: Foster communication between teams
The following practices are encouraged by the guide to foster communication between teams: 1) Use Agile and LEAN practices (e.g. sprint planning and requirements engineering), 2) Collaborate with stakeholders often and provide overall visibility into the project.
This data integration team, implemented the first one, Use Agile and LEAN practices'. However, they only partially implemented the second, Collaborate with stakeholders often and provide overall visibility into the project'. This could indicate the existence of issues in the communication between

teams. The guide would advice the data integration team to first focus on fully implementing the second practice.

Step 2: Foster collaboration between teams
The previous practices already encourage collaboration between teams, however to further meet these expectations, the guide encourages the following practice: Developers get feedback based on releases. This research shows that the data integration team already implemented this practice, meaning that they do not need to change this.

Step 3: Provide testing and maintenance of data products
The following two practices are encouraged by the guide to ensure testing on the data products before releasing to production: 1) Automated and continuous testing in development and staging environment, 2) Use Test driven Development where all code has unit tests.
This research shows that the data integration team already implemented both of these practices.

Furthermore, the following three practices are encouraged by the guide to ensure maintenance of the data products: 1) Make small and continuous releases, 2) Automated and continuous monitoring of applications and resources, 3) Automated dashboards that include health checks and performance.

The last two practices are already implemented by team, both including monitoring activities. However, the first practice Make small and continuous releases', is not defined whether they have implemented it as there were no activities identified around this practice. This guide would advice the team to focus on implementing this practice, if it is not implemented already.

[optionally] Step 4: provide team documentation
Optionally, the guide encourages the implementation of the following practice to ensure team documentation: Integrated configuration management.

This research shows that the data integration team implemented this practice partially which could indicate that the team lacks team documentation. To ensure team documentation, the guide would advice the team to focus on fully implementing this practice.

5.3.2 Unexpected practices

Now that the gaps between expected practices and implemented practices is discussed, the unexpected but implemented practices will be explored. There are four practices are implemented fully and one partially which are not encouraged or expected by the guide. This concerns practices that are about the way in which there is developed. These practices relate to automation and the development process, including the following:

- Practice 1: Automated and continuous deployment throughout entire pipeline.
- Practice 4: Create development sandboxes for minimum code deployment.
- Practice 5: Everything is stored as code and under version control.
- Practice 8: Reduce the time it takes to test, validate and QA code.

- Practice 9: Code reviews are change based.

6 Discussion

This research aimed to design a guide for adopting DevOps practices in a data integration team. The guide for adopting DevOps practices in a data integration team was based on a mapping of expectations to DevOps practices. It guides a data integration team into implementing DevOps practices based on the expectations of stakeholders. To demonstrate the guide, this research assessed the implemented practices and the expected practices from the guide. The development and demonstration of the guide were achieved by answering the following research questions:

1. What are the requirements for a data integration team?
   This research identified fifteen expectations of stakeholders for the data integration team. These expectations indicate the key requirements for a DI team. Mapping these expectations to sixteen DevOps practices from [4, 28], formed the foundation of the guide for adopting DevOps practices in a data integration team. This mapping showed that twelve of the fifteen expectations could be mapped to DevOps practices. Out of the three that could not be mapped, two were related to data management: 'expect high data quality' and 'expect them to be aware of data privacy and compliance.' The remaining one was 'expect to influence cultural change.'

2. Which DevOps practices are currently adopted in a specific data integration team?
   This research identified 48 activities of the data integration team. By mapping the activities to DevOps practices, this research found which practices are currently implemented in the data integration team. Out of the sixteen DevOps practices, ten were implemented by the team, three were partially implemented, and three were not implemented. Furthermore, it showed that the activities on data management could not be mapped to DevOps practices and that the team implemented DevOps practices that are not expected by the guide.

6.1 Key Findings

6.1.1 Stakeholder expectations and Agile methodologies

This research identified expectations of stakeholders of a data integration team. Most of the identified expectations were on communication and collaboration between the data integration team and the stakeholder teams. As discussed in subsubsection 2.1.1, communication, collaboration, and team working is one of the eight components of DevOps. The emphasis on this component within DevOps directly stems from agile methodologies, which are known for the ability to adapt to changing requirements and delivering incremental value through iterative development cycles [7]. Agile frameworks, such as Scrum highlight the importance of collaboration between team members and with stakeholders [33]. Consequently, the guide shows that most of the expectations on communication and collaboration can be achieved by implementing agile practices. Since most expectations could be met by implementing agile practices, it could suggests that agile practices are what the stakeholders seek and not a full DevOps adoption.

6.1.2 Data management gap in DevOps

Despite that this research has shown that most activities of a data integration team could be mapped with DevOps practices, a gap in activities and DevOps practices came to light: the activities

around data management could not be mapped to DevOps practices. The same gap came to light with the expectations of the stakeholders. Their expectations around data management could not be mapped to DevOps practices. These gaps align with [32] and [26], who mention the misalignment of DevOps with data management practices. While DevOps practices can be applied to the infrastructure activities and operational collaboration of a data integration team, they fail to meet the needs of data management within a data integration team, which encompass their main activities.

As the guide is only focused on the implementation of DevOps practices for data integration teams, it is missing expectations and activities for data management. Therefore, to fully support data integration teams, the guide should not be used on it self. Could combining the guide with DataOps practices address this gap?

6.1.3 Including data management with DataOps

As discussed in subsection 2.3, DataOps incorporates components of DevOps, such as automation, quality assurance, and collaboration but it also covers data management activities. Unlike DevOps, which primarily focuses on software development and IT operations, DataOps is tailored to address the unique challenges of managing and processing data. Since DataOps is in line with DevOps while including practices on data management, the guide could be combined with these practices to fulfill the previously discussed gap.

[27] states that DataOps is a general process aimed to shorten the end-to-end data analytic life-cycle time by introducing automation in the data collection, validation, and verification process. It also emphasizes the importance of collaboration between different teams within a data department and the active involvement of all teams in gathering business requirements. Looking back at the expectations reported by stakeholders, expectations on collaboration and communication between teams (within the data department) was reported most, this aligns with the importance of collaboration within DataOps. Both the covering of data management activities and the importance of collaboration within DataOps, indicate that DataOps might be a more suitable approach for data integration teams.

6.1.4 Expanding the guide

This demonstration of the guide showed that there are practices implemented by the data integration team that are not part of the guide. This is due to the foundation of the guide. The guide is based on the mapping of expectations and DevOps practices which makes it only focused on providing practices that have added value to the stakeholders of a data integration team. Which is in line with the question from practice, whether DevOps can address and fulfill specific expectations for a data integration team. Nevertheless, the guide could be expanded with practices that have added value for the data integration team itself. For instance, practices related to automation in the development process were identified to be implemented but not captured by the mapping between expectations and DevOps practices.

Expanding the guide would involve identifying practices that are valuable for the team. This can be achieved by conducting further research, including additional interviews with the data

integration team to identify their needs. These needs can then be mapped with DevOps practices which can help expand the guide with practices based on these needs, making it more beneficial for both the stakeholders and the team itself.

6.2 Future work

This study suggests several recommendations for future work. As mentioned, the guide covers the DevOps practices beneficial for the stakeholders' expectations. Future work could identify the needs of a data integration team and expand the guide with DevOps practices beneficial for these needs. Furthermore, the research showed a misalignment between DevOps practices and the specific data management activities of a data integration team. Future work could research which DataOps practices, which covers elements of DevOps and data management activities, could be applied to data integration teams alongside the guide. Additionally, future work could also research whether DataOps is a better fit for data integration teams overall.

6.3 Limitations

While this research successfully designed and evaluated an guide, several limitations should be acknowledged. These limitations mainly arise due to the constrained methodological choices caused by limited time and resources available for this research.

First, this study was conducted within a single case organization. Which involved interviews with only one data integration team and eight of their stakeholders. Due to this limited scope, the generalizeability of the findings to other organizations or teams is restricted.

Second, the interviews conducted with the DI team to identify activities may not have captured all activities that are currently being used. Additionally, the reported activities may not be fully or accurately applied.

Third, the interviews with stakeholders were based on individual perceptions, which can be biased by personal relationships with the DI team. To limit this bias, at least one representative of each different stakeholder team was interviewed.

Fourth, due to the limited time for this research, the guide was not evaluated externally or implemented in practice. Therefor, the effects of the guide could not be measured and the guide could not be adapted based on practitioners feedback.

Due to the specificity of the case study, the findings of this research cannot be generalized to all data integration teams. They are more applicable within the context of the case organization.

Despite these limitations, this research provides insights which can contribute to a deeper understanding of the impact of DevOps practices and its limitations in data integration teams, both for further studies and organizations.

7 Conclusion

This study examined a data integration team within an e-commerce company of a global manufacturer, aiming to design a guide for adopting DevOps practices in a data integration team. Even though DevOps practices are primarily focused on bridging the gap of development and operations within traditional development teams, non-traditional teams, such as data integration teams, might also experience benefits from adopting DevOps practices. However, emperical evidence on the adoption of DevOps practices within a data integration team could not be found. By mapping expectations for a data integration team to DevOps practices, this study has designed a guide for adopting DevOps practices in a data integration team. This guide was designed based on a question from practice of whether DevOps practices can effectively fulfill specific expectations for a data integration team.

To design the guide, this research started with identifying expectations from stakeholders of the data integration team. Fifteen expectations were identified and mapped to sixteen DevOps practices. This revealed that while twelve expectations could be aligned with DevOps practices, three expectations could not, including the expectations on data management. Further research into the current practices of a data integration team showed that out of sixteen DevOps practices, ten were fully implemented, three were partially implemented, and three were not implemented at all. Notably, activities related to data management did not align with existing DevOps practices, highlighting a gap. The study also found that many of the stakeholder expectations were on communication and collaboration which can be achieved by implementing agile practices. This finding suggests that agile practices might be sufficient to meet many of the expectations without requiring a full DevOps adoption.

While the guide covers most expectations and activities of a data integration team, two limitations came to light. First, the expectations and activities around data management could not be mapped to DevOps practices. This shows the misalignment of DevOps practices and data management activities which align with existing literature that mention the misalignment of DevOps with data management practices. This research looked into DataOps practices to potentially bridge this gap. DataOps incorporates components of DevOps, such as automation, quality assurance, and collaboration but it also covers data management activities. This makes its practices a promising addition to the guide. Thus, data integration team should use this guide to meet stakeholder expectations, but might also need to implement DataOps practices which do cover data management activities. Second, not all practices implemented by the team were covered by the guide, indicating that guide might not align with needs of the data integration team. By understanding the specific needs of a data integration team, this guide can be expanded by additional practices that directly address those needs.

This research concludes that implementing DevOps practices can help data integration teams in fulfilling stakeholders' expectations. However it also raises the question of whether DataOps is more suitable for the data management activities of a data integration team.

References

[1] M. K. Aljundi et al. Tools and practices to enhance devops core values. 2018.

[2] A. Balalaie, A. Heydarnoori, and P. Jamshidi. Microservices architecture enables devops: An experience report on migration to a cloud-native architecture. *Ieee Software*, 33(3):42–52, 2016.

[3] S. Bick, K. Spohrer, R. Hoda, A. Scheerer, and A. Heinzl. Coordination challenges in large-scale software development: a case study of planning misalignment in hybrid settings. *IEEE Transactions on Software Engineering*, 44(10):932–950, 2017.

[4] R. Blinde. Devops unravelled: A study on the effects of practices and technologies on organisational performance, 2022.

[5] A. Brunnert, A. van Hoorn, F. Willnecker, A. Danciu, W. Hasselbring, C. Heger, N. Herbst, P. Jamshidi, R. Jung, J. von Kistowski, A. Koziolek, J. Kroß, S. Spinner, C. Vögele, J. Walter, and A. Wert. Performance-oriented devops: A research agenda. 2015.

[6] I. Bucena and M. Kirikova. Simplifying the devops adoption process. In *BIR Workshops*, 2017.

[7] T. Dingsøyr, S. Nerur, V. Balijepally, and N. Moe. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85:1213–1221, 06 2012.

[8] A. Doan, A. Halevy, and Z. Ives. *Principles of data integration*. Elsevier, 2012.

[9] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano. Devops. *IEEE Software*, 33:94–100, 05 2016.

[10] J. Ereth. Dataops-towards a definition. *LWDA*, 2191:104–112, 2018.

[11] G. B. Ghantous and A. Q. Gill. Devops: Concepts, practices, tools, benefits and challenges. In *Pacific Asia Conference on Information Systems*, 2017.

[12] J. R. Hamilton et al. On designing and deploying internet-scale services. In *LISA*, volume 18, pages 1–18, 2007.

[13] A. Hashmi, S. Ishak, and H. B. Hassan. Role of team size as a contextual variable for the relationship of transformational leadership and teamwork quality. *Asian Journal of Multidisciplinary Studies*, 6(5):76–81, 2018.

[14] A. Hemon-Hildgen, B. Lyonnet, F. Rowe, and B. Fitzgerald. From agile to devops: Smart skills and collaborations. *Information Systems Frontiers*, 22, 08 2020.

[15] A. Hermawan and L. P. Manik. The effect of devops implementation on teamwork quality in software development. *Journal of Information Systems Engineering and Business Intelligence*, 7(1):84, 2021.

[16] A. Hevner and S. Chatterjee. *Design research in information systems: theory and practice*, volume 22. Springer Science & Business Media, 2010.

[17] J. Humble and G. Kim. *Accelerate: The science of lean software and devops: Building and scaling high performing technology organizations*. IT Revolution, 2018.

[18] J. Iden and B. Bygstad. The social interaction of developers and it operations staff in software development projects. *International Journal of Project Management*, 36(3):485–497, 2018.

[19] J. Iden, B. Tessem, and T. Päivärinta. Problems in the interplay of development and it operations in system development projects: A delphi study of norwegian it experts. *Information and Software Technology*, 53(4):394–406, 2011.

[20] R. Jabbari, N. Ali, K. Petersen, and B. Tanveer. What is devops?: A systematic mapping study on definitions and practices. pages 1–11, 05 2016.

[21] T. Kude, S. Mithas, C. T. Schmidt, and A. Heinzl. How pair programming influences team performance: The role of backup behavior, shared mental models, and task novelty. *Information Systems Research*, 30(4):1145–1163, 2019.

[22] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles. A survey of devops concepts and challenges. *ACM Comput. Surv.*, 52(6), nov 2019.

[23] M. Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246, 2002.

[24] L. E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä, J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo, and C. Lassenius. Devops in practice: A multiple case study of five companies. *Information and Software Technology*, 114, 06 2019.

[25] L. E. Lwakatare, P. Kuvaja, and M. Oivo. An exploratory study of devops: Extending the dimensions of devops with practices. 08 2016.

[26] K. Mainali, L. Ehrlinger, M. Matskin, and J. Himmelbauer. Discovering dataops: a comprehensive review of definitions, use cases, and tools. In *DATA ANALYTICS 2021 The Tenth International Conference on Data Analytics*, 2021.

[27] A. R. Munappy, D. I. Mattos, J. Bosch, H. H. Olsson, and A. Dakkak. From ad-hoc data analytics to dataops. In *Proceedings of the International Conference on Software and System Processes*, ICSSP '20, page 165–174, New York, NY, USA, 2020. Association for Computing Machinery.

[28] T. Offerman, R. Blinde, C. J. Stettina, and J. Visser. A study of adoption and effects of devops practices, 2022.

[29] A. Palmer. From devops to dataops - dataops tools transformation — tamr. `https://www.tamr.com/blog/from-devops-to-dataops-by-andy-palmer/`, 2015. Retrieved: 2024-03-28.

[30] C. Patel and M. Ramachandran. Agile maturity model (amm): A software process improvement framework for agile software development practices. *International Journal of Software Engineering, IJSE*, 2(1):3–28, 2009.

[31] K. Peffers, T. Tuunanen, C. E. Gengler, M. Rossi, W. Hui, V. Virtanen, and J. Bragge. Design science research process: A model for producing and presenting information systems research. *arXiv preprint arXiv:2006.02763*, 2020.

[32] M. Rodriguez, L. J. P. de Araújo, and M. Mazzara. Good practices for the adoption of dataops in the software industry. *Journal of Physics: Conference Series*, 1694(1):012032, dec 2020.

[33] K. Schwaber and J. Sutherland. The scrum guide. *Scrum Alliance*, 21(1):1–38, 2011.

[34] M. Senapathi, J. Buchan, and H. Osman. Devops capabilities, practices, and challenges: Insights from a case study. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, EASE'18. ACM, June 2018.

[35] M. Senapathi, J. Buchan, and H. Osman. Devops capabilities, practices, and challenges: Insights from a case study. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pages 57–67, 2018.

[36] M. Shahin, M. A. Babar, and L. Zhu. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. *IEEE access*, 5:3909–3943, 2017.

[37] D. Stahl, T. Martensson, and J. Bosch. Continuous practices and devops: Beyond the buzz, what does it all mean? In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 440–448. IEEE, 2017.

[38] B. Tessem and J. Iden. Cooperation between developers and operations in software engineering projects. In *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pages 105–108, 2008.

[39] A. Thusoo and J. Sarma. *Creating a Data-Driven Enterprise with DataOps*. O'Reilly Media, Incorporated, 2017.

[40] A. Wiedemann, M. Wiesche, H. Gewald, and H. Krcmar. Understanding how devops aligns development and operations: a tripartite model of intra-it alignment. *European Journal of Information Systems*, 29(5):458–473, 2020.

[41] R. T. Yarlagadda. Understanding devops & bridging the gap from continuous integration to continuous delivery. *Understanding DevOps & Bridging the Gap from Continuous Integration to Continuous Delivery', International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN*, pages 2349–5162, 2018.

[42] H. Yasar and K. Kontostathis. Where to integrate security practices on devops platform. *International Journal of Secure Software Engineering*, 7:39–50, 10 2016.

A Interview Guide Stakeholders

Interview Questions Clients

Introduction

Thank you for joining me today. I'm conducting research for my Bachelor's thesis in Computer Science & Economics about the working method of a data integration team. The goal is to enhance data integration team practices based on stakeholder perspectives. Today, I'm interested in gathering your insights on collaboration and expectations related to data integration teams.

I would like to remind you that your participation is confidential, and all information shared will be anonymised for research purposes.

Thank you for your participation. Let's begin the interview.

Background

- Could you introduce yourself?
- Could you describe the role of your team within the organization?
- Could you describe your role towards the data integration team?
- How long have you been working with data integration teams?

Topic

- **Collaboration with data integration**
  - Could you describe how your teams collaborates with the data integration team?
    - What is the output your team receives from the data integration team?
    - What is the input your team delivers to the data integration team?
    - What do you do with the output that the data integration team delivers?
    - Could you describe how you communicate with the data integration team?
- **Expectations of data integration**
  - What are your expectations regarding communication with the data integration team?
  - What are your expectations regarding collaboration with the data integration team?
  - What do you expect regarding quality of the output?
- **Current perception of expectations**
  - What is your current perception of the communication with the data integration team?
  - What is your current perception of the collaboration with the data integration team?
  - What is your current perception of the quality of the output?

- **Expectation of data integration teams outside of the current organization**
  - Based on experiences prior to this organization, what would other expectations be for a data integration team?
  - Do you have any other comments/remarks/questions?

B Interview Guide Team

Interview Questions Team

Background

- Could you describe your role within the data integration team?
- How much experience do you have with DevOps?
- Could you elaborate on what your experiences with DevOps are?

Topic

- **Development & Operations**
  - Could you describe what can be seen as development and operations within your team?
  - Could you describe what your role is in development and operations within your team?
  - Could you elaborate on how development and operations work together within your team?
- **Continuous integration & Continuous deployment**
  - Could you elaborate on which activities your team uses for continuous integration?
  - Could you elaborate on which activities your team uses for continuous deployment?
  - Could you elaborate on which activities your team uses for automation?
  - Could you elaborate on which activities your team uses for quality assurance?
- **Communication, Collaboration, Team working**
  - Could you elaborate on which Agile and/or LEAN activities your team uses?
  - Could you elaborate on which activities your team uses to promote communication among team members?
  - Could you elaborate on which activities your team uses to promote collaboration among team members?
  - Could you elaborate on which activities your team uses to promote communication among other departments or teams?

- Could you elaborate on which activities your team uses to promote collaboration among other departments or teams?

- Could you elaborate on which activities your team uses to ensure effective team-working?

- Could you elaborate on which activities your team uses to promote knowledge sharing?

C Mapping Activities and DevOps practices

A mapping of activities reported by the DI team to DevOps practices from [4, 28].

	1 Automated and continuous deployment throughout entire pipeline	2 Make small and continuous releases	3 Developers get feedback based on releases	4 Create development sandboxes for minimum code deployment	5 Everything is stored as code and under version control	6 Integrated configuration management	7 Automated and continuous testing in development and staging environment	8 Reduce the time it takes to test, validate, and QA code	9 Code reviews are changed based	10 Automated and continuous monitoring of applications and resources	11 Automated dashboards that include health checks and performance	12 Support configurable logging that can optionally be turned on/off as needed	13 Use trunk-based development over long-lived feature branches	14 Use Test driven Development where all code has unit tests	15 Use Agile and LEAN practices (e.g. sprint planning and requirements engineering)	16 Collaborate with stakeholders often and provide overall visibility into the project
Relational and primary key validation																
Post-transformation data validation																
Basic Null and integrity checks																
Consistency checks																
source validation																
Data quality checks																
Automated data refreshing																
Automated data ingestion																
Data modeling standardization																
Data processing pipeline																
Updating and deploying seed files	M															
Sprint reviews															X	X
Daily check															X	
Retrospective meetings															X	
Demo sessions			X												X	X
Pair programming															X	
2 week sprints															X	
Architecture and design discussions															X	
User documentation															X	
Team documentation						X									X	
Unit testing							X							X		
Automated tests on code							X									
Automated validation							X									
System integration testing							X									
Development environment testing				X												
Stakeholder validation			X													
User acceptance testing			X													
Cluster logging						X				X						
Failed test alerts										X						
Automated alerts										X						
Failing pipeline alerts										X						
Automated Notification on Issues										X						
Workflow dashboards											X					
Progress monitoring											X					
Repositories					X	X										
Version Control					X	X										
Code review									X							
Pull requests						X										
Peer reviews on pull requests									X							
User stories																
Biweekly Alignment sessions															X	X
Weekly meeting																X
Weekly Scrum of Scrums															X	X
Workshops															X	
Ticket registration															X	
Automated deployment pipeline	M															
Development pipelines								X								
Testing pipelines								X								