CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Exploring the Criticality and Impact of DevOps Practices

Master's thesis in Software Engineering

MARTIN ARVEDAHL
CHRISTOPHER ÅKERSTEN

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY AND UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Master's thesis 2018

Exploring the Criticality and Impact of DevOps Practices

MARTIN ARVEDAHL

CHRISTOPHER ÅKERSTEN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology and University of Gothenburg
Gothenburg, Sweden 2018

Exploring the Criticality and Impact of DevOps Practices

MARTIN ARVEDAHL
CHRISTOPHER ÅKERSTEN

Supervisor: ERIC KNAUSS, Department of Computer Science and Engineering
Examiner: JAN-PHILIPP STEGHÖFER, Department of Computer Science and Engineering

Master's Thesis 2018
Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Word cloud based on the interview transcripts.

Typeset in LaTeX
Gothenburg, Sweden 2018

Exploring the Criticality and Impact of DevOps Practices
MARTIN ARVEDAHL
CHRISTOPHER ÅKERSTEN
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

The purpose of the study has been to explore DevOps practices with regards to their criticality on DevOps adoption goals, making grounds for decision making in the industry and for further research. Through performing a qualitative multiple case study on practitioners of DevOps, with existing literature as input, a set of adoption goals, practices and effects was elicited. The multiple case study consisted of interviewing and surveying twelve practitioners of DevOps in various software organizations, with different levels of experience. We have found that there seems to be a set of practices shared among practitioners, such as Automatic Testing & System Monitoring, critical for a set of shared adoption goals like Throughput. Some of these practices are transforming the way of working in the industry with regards to organizing operations. We further found that the implementation of the practices seems to have a significant effect on considered adoption goals, motivating further research. Lastly, we found that practitioners in the industry are not operationalizing DevOps, often not measuring the effects of adoption. However, some suggested metrics for verifying the effect of implementing DevOps practices have been found.

Keywords: DevOps, Goals, Practices, Effects

Acknowledgements

Thanks to Eric Knauss, our supervisor, for assistance and advice. This thesis would not be what it is without your input. Thanks to Peter Lindh and Fredrik Normén, our contact persons at Squeed AB, for letting us drink your coffee, receiving your feedback and providing us with a valuable network of people to interview. Lastly, we extend thanks to all practitioners who have participated in our thesis work providing their valuable expertise. Additionally, we would like to thank the examiner, Jan-Philipp Steghöfer, for valuable feedback and well-thought-out criticism.

Martin Arvedahl and Christopher Åkersten, Gothenburg, June 2018

Contents

1

Introduction

The term DevOps, short hand for Development Operations [13, p. 5] was coined in the sphere of software engineering to bridge the gap between Development and Operations to better facilitate fast delivery of software. As agile development methodologies gained ground in the recent decades affecting the development process and culture, the operational aspects of software have in many cases been left behind, making this gap apparent. Traditionally, the development team and the operations team have been isolated from each other. When the developers have finished coding, they have simply handed over the code to the operations team, and have then forgotten about it [2, p. 10].

DevOps is nowadays widely considered by the industry but remains poorly understood in academic literature. While the term DevOps has continually increased in popularity since it was coined in 2009 [32], there is a lack of agreement on the goals, activities performed, and effects within the academic literature. In particular, the effects of adopting what might be considered DevOps practices has barely been explored at all.

The goals of DevOps are not clearly defined in the academic literature. The generalized definition seems to be to increase the cooperation between Dev and Ops [13]. This would however seem to be a means to an end. Some practitioners are clearer on that improving collaboration is only a subgoal. The end goals vary, where increased throughput of delivery is generally agreed upon as one of the goals. This ambiguity shows the importance of understanding to what end DevOps is considered by practitioners in the industry.

When talking about implementing DevOps, a set of development practices and cultural shifts are usually considered [13, p. 5]. There is however not a consensus in the literature on what DevOps is [13, p. 6], and hence what practices are included. While there are claims to the contrary outside of the academic literature that states that DevOps is mature and well defined such as the State of DevOps Report [11, p. 4], the academic literature remains without consensus.

Looking at the effects of implementing DevOps, finding studies covering any kind of measurements of the effect is difficult. One exception is the State of DevOps Report connecting the performance of software teams with different practices [11, p. 19]. It is however limited in scope with regards to what practices are considered, compared to the practices presented in the literature. Furthermore, it remains unclear to what

degree practices contribute to common adoption goals. Additionally, their data is private property and they mainly present their positive findings. Qualitative studies have been performed to study the effects of DevOps practices. One example is a qualitative study by Erich et al. where exploratory interviews were performed on six companies with the hopes of deriving quantitative measures for the effectiveness of DevOps [5]. They claim that none of the interviewees shared measurements and propose this as an area of future research [5, p. 15].

To show that the implementation of DevOps is beneficial, there is a need to understand why DevOps is considered, what practices are implemented and how critical these practices are. Currently, the academic knowledge surrounding DevOps seems limited. The purpose of this study is first to propose a set of adoption goals when implementing DevOps. Second, to propose a set of practices that should be considered when adopting DevOps. Third, to propose how critical these practices may be for the adoption goals. Fourth, to make ground for further research validating the criticality of the proposed practices for reaching adoption goals. The latter involves suggesting effects of the practices and finding metrics that could be used in further studies. Hence, we derived the following research questions:

RQ1 *What are goals of DevOps adoption?*

The definition of DevOps is unclear, and thus the goals of adoption are poorly understood. In order to understand for what purpose practices are implemented, there is a need to understand the adoption goals with DevOps. This is an exploratory question: What are the goals of DevOps adoption?, and is thus suitably answered through qualitative data [4, p. 287]. In this study it is approached through existing academic literature and qualitative data from interviews with a complementing survey for verification.

RQ2 *What practices fall within the scope of DevOps?*

With an unclear definition of DevOps, there is a need to explore what practices are considered when adopting DevOps. This is an exploratory question: What DevOps practices exists? In this study, it is approached by using existing academic literature and qualitative data from interviews.

RQ3 *What is the criticality of DevOps practices for the adoption goals?*

For the practices that emerge, there is gain in attempting to classify the criticality of the practices for the adoption goals. This is a classification question: What is the criticality of the practice(s)? In this study, this question will not be answered in its entirety, but mainly the study will make way for further research by proposing what practices seems to be critical, so they can be verified in future research. This will be done through asking practitioners what practices they consider critical for their adoption goals. This will mainly be done with a survey that is constructed based on the practices and goals that emerge from the interviews.

RQ4 *What is the perceived impact of implementing DevOps practices?*

From an understanding of what practices DevOps consists of, it is interesting to note what impact adopting different DevOps practices has, apart from how important they have been for the adoption goals. This is an exploratory question: What is the perceived impact of implementing DevOps practices? This question will be considered during the interviews, where perceptions of commonly observed effects will be elicited. It is expected that effects outside of the adoption goals will emerge, positive and negative. Moreover, the survey will complement these findings with perceived effects in relation to the adoption goals.

RQ5 *To what extent can the impact of adopting DevOps practices be measured?*

To enable quantification of impact in future research, exploring how practitioners either measure or suggest measuring the impact of adopting DevOps practices is relevant. This is an exploratory question: How can we measure it? By asking practitioners how they have measured the impact as well as how they propose it can be measured, the emerging metrics can be useful in future research.

With respect to the review of existing literature described in Chapter 2, there are mainly two groups that can make use of this study, researchers and practitioners. For researchers, this study mainly provides knowledge. It is an entry point of how the effect of DevOps practices can be measured. There exist few guidelines to show what positive and negative impact different DevOps practices have. By beginning to address this gap, we enable future empirical research to show the benefits and/or drawbacks of DevOps through for example experimental studies. This study can also serve as an entry point with regards to gathering quantitative data of DevOps practices to show their benefits and/or drawbacks. Such future research would aid the industry in decision making with regards to adopting DevOps practices. For the practitioners, mainly DevOps consultants or similar, this study provides some common goals for DevOps adoption and gathered experience about what practices to consider. The study also provides an entry point for showing the main benefits and/or drawbacks of implementing DevOps.

2

Related Work

We initially looked at SLRs and mapping studies defining DevOps. Among these, we found a Systematic Mapping Study by Jabbari et al. [13] as well as a Systematic Literature Review by Erich et al. [6]. They state that when implementing DevOps, a set of development practices and cultural shifts are usually considered such as continuous delivery and automated deployment [13, p. 5]. In the systematic mapping study by Jabbari et al. the following definition was derived: *"DevOps is a development methodology aimed at bridging the gap between Development (Dev) and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices."* [13, p. 8]. It would seem like the literature suggest that DevOps has two parts, cultural values and technical practices respectively. There is however not a consensus in the literature on the definition of DevOps and what practices are included [13, p. 6]. While there exist claims to the contrary to be considered that states that DevOps is mature and well defined such as the State of DevOps Report [11, p. 4], that is not the case in academic literature, but rather claimed by blogs and white papers.

Looking at various blog posts and white papers, practitioners have defined acronyms compromised of core values that describes what DevOps is. Some examples are:

- CAMS (Culture, Automation, Measurement, Sharing) [32]
- CALMS (Culture, Automation, Lean, Measurement, Sharing) [32]
- CAMM (Collaboration, Automation, Measurement, Monitoring) [21, p. 214]

These acronyms serve well in explaining the bigger picture of what adopting DevOps involves. Though, each term would need to be explored further since they are quite broad. A brief analysis shows that Culture and Collaboration seems to be used interchangeably, being about creating a culture of collaboration between Dev and Ops and is included in all definitions found. The other terms vary, some emphasize knowledge sharing while others do not, and so on.

After looking at defining DevOps, we investigated the goals of adopting DevOps. Jabbari states that the main goal of DevOps is to bridge the gap between development and operations [13, p. 5]. Bridging the gap is, however, only considered a subgoal that is not fulfilling anything in itself. Patrick Debois — founder of *devopsdays* [3] — states that DevOps originated in that *"Small groups of both developers and operations people delivered new releases and understood that they were*

working toward the same goals for the same customers." [2, p. 5], thus putting more focus on the customers than the process in itself. Gene Kim — practitioner, researcher and author on the topic — states that *"The goal of DevOps is not just to increase the rate of change, but to successfully deploy features into production without causing chaos and disrupting other services, while quickly detecting and correcting incidents when they occur."* [14, p. 6], which puts more emphasis on the outcome rather than the process. Similarly, Erich et al. cite that *"The goal of DevOps has been defined as that of reducing the time between development and operation of software without negatively affecting quality."* [5, p. 1]. Looking at the State of DevOps Report 2017, various other goals are mentioned as well, such as the quality of products and services and operating efficiency [11, pp. 30-31]. With this variety in mind, it is interesting to understand what goals practitioners tend to focus on when adopting DevOps.

Next, we explored what literature existed with regards to the effects of DevOps practices. Very little research was found for the effect of DevOps, especially with regards to quantitative data. A very recent qualitative study by Erich et al. [5] made an attempt at gathering quantitative measurements, among other things, through exploratory interviews. This was done unsuccessfully, and they state that the field is open for more research in quantitative measurements and gathering quantitative data to show the benefits of DevOps. They state that the organizations they interviewed were too early in their stages of adoption, and that they only interviewed a handful of companies. By exploring more companies, in later stages, one may get different results.

Further, the State of DevOps Reports, which are a set of survey-based studies made by a set of corporate organizations, imply that they have metrics as well as quantitative data that prove the benefits of DevOps [11, 10, 26, 8]. There exists little other academic support to their claims, but the metrics they have defined might be of value. They have chosen to define IT performance profiles based on clustering on the following metrics [10, p. 15]:

- Mean Time To Recover (MTTR) — Time it takes to restore from a service incident for example.
- Lead time for changes — Time from code commit to being live in production.
- Change failure rate — How big percentage of the changes that lead to a "degraded service or subsequently require remediation".
- Deployment frequency — How often deployment occurs.

They have then connected these profiles to different practices and such claimed some cause-effect relationships. However, the related peer-reviewed literature is significantly more conservative with regards to the cause-effect relationships between practices and measurements [9] than the white papers. They claim to have followed a grounded theory approach in the DevOps report, where they reach a model for a set of practices and outcomes that they claim are statistically significant [11, p. 50][11, p. 19]. What remains unclear is the process of selecting a set of practices. The set of practices in their model does not align completely with the other academic research available with regards to practices considered for DevOps. Furthermore,

they do not show the level of impact the different practices have. There is also no insight into their data as it is their private property. It would serve well to explore the practices and goals considered by the DevOps practitioners themselves to verify and/or complement this research.

To find methods to approach the constructed research questions, we looked in to empirical methods for software engineering research. Different types of methods are presented in the literature for different uses [4, p. 286], and arguing that what method to use is highly connected to the type of research question we are asking [4, p. 287]. This opened for exploring different research methodologies for qualitative data gathering in the literature [24] as well as ways of interpreting qualitative data [31].

3

Methodology

In the following chapter, the different methods used to answer the research questions are presented. It starts describing the procedure of the Literature Review in order to derive DevOps practices and adoption goals, followed by a more in detail description of the Qualitative Multiple Case Study consisting of semi-structured interviews and a complementing survey. The process of selecting the cases, preparing the interviews and conducting them, as well as analyzing the qualitative data through thematic coding is described. Finally, the process of triangulation through the complementing survey is laid out. In Table 3.1 the mapping between the research questions and the methodologies used is shown.

Table 3.1: A mapping between the research questions and the methodologies

Research Question	Data Sources
RQ1: What are goals of DevOps adoption?	Literature; Interviews; Survey
RQ2: What practices fall within the scope of DevOps?	Literature; Interviews; Survey
RQ3: What is the criticality of DevOps practices for the adoption goals?	Interviews; Survey
RQ4: What is the perceived impact of implementing DevOps practices?	Interviews; Survey
RQ5: To what extent can the impact of adopting DevOps practices be measured?	Interviews; Survey

3.1 Literature Review

To understand what exists in the body of knowledge related to DevOps goals and practices, the literature was reviewed. While searching for DevOps, a recent Systematic Literature Review and a Systemic Mapping Study in the field of DevOps were found [6, 13]. Due to this, there was no reason to conduct an additional one on defining DevOps in particular. With the starting point in these papers, goals and practices of DevOps was sought for in order to provide input into the multiple case study.

3.2 Qualitative Multiple Case Study

The main data gathering approach was a qualitative multiple case study with semi-structured interviews at its core, where a selection of practitioners of DevOps were

the cases. The aim of the interviews was to answer **RQ1–5** by deriving goals and practices central for DevOps, understand what practices are potentially critical for reaching certain goals as well as find out the perceived impact of DevOps practices and to what extent that impact can be measured. The data from the interviews prepared the grounds for the complementing survey for validating and complementing the results.

3.2.1 Study Design Rationale

The reason for doing a multiple case study was partly due to the nature of DevOps being a loosely defined concept, and as such an exploratory approach was fitting. According to Yin, a case study is defined as *"an empirical inquiry that investigates a contemporary phenomenon within its real-life context; when the boundaries between phenomenon and context are not clearly evident; and in which multiple sources of evidence are used."* [33, p. 18]. With multiple cases there would further be a greater chance of higher validity [4, p. 297]. With this is mind, a multiple case study could assist in providing answers to the research questions. According to Runeson et al., case studies *"provide a deeper understanding of the phenomena under study"* [28, p. 4] even though they *"do not generate the same results on, for example, causal relationships, as controlled experiments do"* [28, p. 4]. To sum up, the multiple case study approach is a good fit to the research questions since they have an exploratory nature.

3.2.2 Case Selection

In this study a case is a practitioner of DevOps, in other words an industry expert on DevOps responsible for adoption. The entry point for the case selection was the network of practitioners provided by Squeed AB. Their network covers many industries and is broad. The practitioners contacted within their network in turn led to additional contacts and so on, leading to a snowball sampling method with regards to practitioners. The practitioners were contacted by email, phone, chat and in person. The sampling of cases was done with the main criterion that the selected practitioners should consider themselves as being an "implementer of DevOps". The practitioners consequently had for the most part a coaching role, with various levels of technical background, where they have overseen a DevOps adoption in one or several organizations on behalf of either the adopting organizations or a specialized consultancy firm. The practitioners are neither Dev or Ops but rather have a strategic role overseeing both. The set of practitioners can be seen in Table 3.2. The table further shows the experience level of the practitioners. Noting that all are practitioners of DevOps, and thus in some regard experts, the experience within the case selection still varies. The experience level was elicited in a later phase after analysing the interview data and is based on how long the practitioners have worked with DevOps, the maturity of the contexts they have been working in, as well as their self-perceived level of experience. The sampling of cases was limited

to the geographical context of the study, the maturity of DevOps adoption in the geographical context, the contact network as well as the availability of practitioners.

The reason that twelve cases were selected was that themes started to reoccur such that more general conclusions about the results could be drawn, as well as there being little new data emerging from the interviews.

Table 3.2: The practitioners of DevOps selected, with their role, employment, adoption contexts and estimated level of experience.

Practitioner	Role	Employed at	Adoption Context	DevOps Experience
A	DevOps Consultant	ConsultancyFirm1	RetailComp1	Very High
B	DevOps Consultant	ConsultancyFirm2	Various	Very High
C	DevOps Consultant	ConsultancyFirm3	Various	Very High
D	Agile Coach	ConsultancyFirm1	Various	Mid
E	Agile Coach	ConsultancyFirm1	AutomotiveComp1	High
F	Agile Coach	ConsultancyFirm1	AutomotiveComp1	Mid
G	Agile Coach	ConsultancyFirm4	AutomotiveComp1	Mid
I	Technical Agile Coach	DevComp1	DevComp1	Very High
J	Enterprise Architect	FinComp1	FinComp1	Very High
K	System Architect	ConsultancyFirm5	LeasureComp1	High
H	CTO	RetailComp1	RetailComp1	Mid
L	Platform Area Business Analyst	AutomotiveComp2	AutomotiveComp2	Very High

3.2.3 Semi-structured Interviews

The use of semi-structured interviews was selected due to the capability of both validating the literature review with structured questions, as well as opening up for additional exploratory questioning [23, p. 4]. The literature review led to a set of suggested practices and adoption goals that needed verification, but there could also have been any amount of practices and adoption goals not considered in the literature, but relevant according to the practitioners. As such, there was a need for open-ended questions as well, which semi-structured interviews offer [29, p. 14]. Additionally, the purpose of the interviews was to derive the perceived criticality of the practices, impacts of practices, and metrics and measurements for the impact of the practices. Since the literature did not offer much in that regard, open-ended exploratory questions were crucial.

Interview Guide

An interview guide was constructed to support the semi-structured interview found in Appendix A with questions mapping to the different research questions. While constructing the interview guide, aid from some guidelines from Harvard University [12] were used. The step-by-step guide to formulate interview questions was especially helpful [12, p. 4]. The structure was impacted by the guide, where warm-up and closure questions were used. The guidelines further helped in setting the tone of the questions, focusing on eliciting the experience and expertise of the in-

terviewees letting them tell their 'story'. Lastly, the guidelines offed some dos and don'ts such as ask 'how' over 'why'.

The interview guide was divided into four separate sections of questions, where each question was connected to a corresponding research question if applicable, as well as a short rationale. The first set of questions are intended to function as warm up questions to get the interviewee going. Further, they set the context of the particular case such as the role of the practitioner, the type of company and kind of product(s) the practitioner has worked on and so on. The second section focuses on questions on defining DevOps practices. Worth noting is that examples of DevOps practice areas from the literature were used to guide the interviewees, as can be seen in Appendix A. These were primarily taken from work by Erich et al. etc. [5, p. 15][32] and were presented after asking exploratory questions with regards to DevOps practices. This was done in order to verify the literature, but also in some cases to help the interviewees get going, where some exemplification helped. The third section focuses on the operationalization of DevOps. The questions are related to measurement and metrics, with further specific measurements in a few key areas based on existing research. Lastly, the fourth section of the guide ends with some wind down questions to wrap up and finish the interview in a good mood.

Conducting the Interviews

The interviews were conducted with the practitioners together with the thesis writers. Each interview was recorded with permission from the interviewees and they were promised anonymity. The interviews took between 40 and 80 minutes to perform, and during the interview the interview guide was followed. Since the interview was semi-structured, it left a lot of room for sidetracking into different stories and areas. The practitioners got a version of the interview guide to make sure they could follow the progress to manage the time constraints. The suggested practices and measurement areas where omitted from their version of the interview guide to prevent a biased response. After the interviews, the recordings were transcribed. These are not publicly available though, due to the anonymity of the interviewees. All interviews were held in during the period March to April 2018.

Interpreting the Interview Data

By using the program NVivo 11 [27], each interview was transcribed, and then a process of thematic coding ensued. This was useful to *"highlight priorities and provide focus to the process of analyzing qualitative data"* [31, p. 50]. By doing this, different concepts could be categorized, and it would be easier to overview the data and find what different people said about a certain concept (like a DevOps practice or an adoption consequence). The approach was to let the thematic codes emerge from the data and hence build a set of categories, as is one of the suggested approaches by Creswell [1, p. 187]. As such, answers could for example be connected to certain DevOps practices or consequences.

The thematic coding was done over two iterations. After the first round of thematic coding the codes were hierarchically grouped into broader categories such as "DevOps practices". The result of the hierarchical grouping can be found in Appendix B. The second round of coding was performed to make sure that the later emerging codes were similarly covered in the earlier thematically coded sources. By having both thesis writers code each source at least once the validity of the coding increased and the ambiguity of the codes decreased.

3.2.4 Survey

To verify and to complement the findings from the interviews, a survey was made and was sent to practitioners with the practices and goals elicited as input. This prevented errors with practices being forgotten during the interviews for example. The response rate for the survey was 100%, including answers from all the twelve interviewed practitioners.

To properly construct the survey, some advice from academic literature was consulted. Since making a survey is not just about making a questionnaire, but *"a comprehensive system for collecting information to describe, compare or explain knowledge, attitudes and behavior"* [25, p. 16], it was useful to follow a process, in this case the one by Kitchenham and Pfleeger. They have published a guide for constructing surveys in six parts, each in a separate publication [25, 15, 16, 17, 18, 19]. Some of the advice, however, seem to be more relevant with regards to quantitative research. As such, they were not taken into consideration. In addition, since this qualitative survey was aimed to the interviewed practitioners, terms like sample size, thorough data analysis, etc. was not considered, while being considered by the authors in their last two parts of the guide [18, 19].

The process was especially helpful when creating the questionnaire to make the questions well-formulated and purposeful [16, p. 21] and avoiding pitfalls such as *"avoid the use of abbreviations, slang and colloquial expressions"* [16, p. 22] or using negative statements [16, p. 22]. With regards to the response format, the authors suggest using an ordinal standardized response format throughout the survey (Strongly agree to Strongly disagree) [16, p. 21], which was followed. Moreover, there were both a neutral choice and a "No opinion"/"I don't know" option to prevent the respondents from feeling forced to answer something as per their recommendations [16, p. 22]. At last, the questions were discussed with the academic supervisor to see if the questions were relevant and comprehensible [17, p. 21] and piloted on a person, who is not part of the study, to check for ambiguities.

The questionnaire itself was made using SurveyMonkey [30], and the questions and corresponding response options can be found in Appendix C. A link to the survey was emailed out to the practitioners, with an instruction and a time estimate included, as is recommended [16, p. 23].

4

Results

This chapter presents the findings related to the research questions. The data presented originates from the literature review, interviews and survey and is further discussed in detail in Chapter 5.

4.1 RQ1: *What are goals of DevOps adoption?*

In the interviews, goals for DevOps adoption were explored, since the results from the literature review was inadequate, as can be seen in Chapter 2. The transcriptions from the interviews were thematically coded with the code "Goals of Implementing DevOps". Then, all the referenced data to that code was queried and further categorized in broader goal categories. These are presented in Table 4.1. The quotes in the table have been translated from Swedish to English and exemplify instances of where the goals were found. Looking at the interview data for each practitioner individually, what goals were mentioned as relevant can be seen in Table 4.2. These goals were used as part of the input to the survey and have been sorted based on the scores from the survey results.

Table 4.1: The DevOps adoption goals elicited from the interviews, with a few quotes to exemplify the underlying data. The quotes have been translated from Swedish to English.

Goal	Description
Throughput	Faster Delivery, Reduce time to market. *"We want shorter time to market. We must reduce the cycle times."* — Practitioner I
Value of Delivery	Deliver the right thing. *"What is the value of our delivery? The value can be sh*t. I can deliver any number of features without generating value. So, it is absolutely about this, we must find the value."* — Practitioner H

...

Goal	Description
Product Quality	Stability, Availability, Maintainability, ... *"You do not have to make a trade-off between stability and throughput. High-performers are better at both"* — Practitioner C *"Increasing the quality is important. The faster you release features of the best possible quality, the better trust you gain with the market and win over your competitors"* — Practitioner A
Work Environment	Remove tedious tasks, Reduce deployment pain, Employee Satisfaction... *"[...] from the old ways, you sit as a developer in a large company, build something and when you are done with the coding you hand it over to the ones who will maintain it. You have a lot of hassle, tedious tasks with unnecessary documentation, and then writing papers and tests back and forth. That is slow, tedious, and people get annoyed at each other."* — Practitioner J
Product Scalability	The capacity to scale up the product.
Organization Scalability	The capacity to scale up the organization.
Reduce Truck Factor	Reduce the dependency on key people.

Table 4.2: DevOps adoption goals mentioned as important in the interviews with each practitioner, where (×) means it is mentioned as important.

Goal	A	B	C	D	E	F	G	H	I	J	L	K
Throughput	×	×	×	×	×	×	×	×	×	×	×	×
Product Quality	×	×	×	×		×			×		×	×
Value of Delivery							×	×				
Work Environment										×		×
Product Scalability											×	
Organization Scalability		×										
Reduce Truck Factor		×										×

One of the purposes of the survey was to validate the interview data with regards to what goals practitioners have considered when adopting DevOps. The results from the survey can be seen in Table 4.3. All interviewees who mentions the goal in the interview also considers it in the survey, except in one case, which is practitioner D regarding Product Quality. There are in addition many, who affirm adoption goals in the survey, that did not mention them in their interviews. Looking at the average

score of all practitioners, all goals are above the "neutral" stance on average. One thing that was noted in the interviews was that the experience of the practitioners regarding DevOps adoptions varied. By grouping the results by level of experience, a weighted average was calculated that can be seen in the W.Avg column. The weighted average shows an increased importance for all adoption goals apart from Throughput, were Work Environment, Product Scalability, Organization Scalability and Reduce Truck Factor all greatly increased compared to the average.

Table 4.3: DevOps adoption goals ranked by importance by practitioners using an ordinal scale from strongly agree to strongly disagree, here represented as numbers ranging from 2 to −2. Here, the practitioners are grouped by the level of experience ranking from very high to mid, which can be seen in Table 3.2. The weighted average is then presented where a practitioner with a very high level of experience with DevOps has Weight 3, high has Weight 2 and mid has Weight 1.

DevOps Experience	Very High (3)						High (2)		Mid (1)				Avg	W. Avg
Goal	A	B	C	I	J	L	E	K	D	F	G	H		
Throughput	2	2	1	1	2	2	2	2	2	1	2	2	1.75	1.73
Product Quality	2	2	1	1	2	1	2	2	-1	1	1	1	1.25	1.42
Value of Delivery	2	2	0	0	2	1	1	2	1	0	1	1	1.08	1.15
Work Environment	2	2	1	1	2	1	0	1	-2	0	-2	1	0.58	1.00
Product Scalability	2	1	2	1	2	2	0	0	-2	-1	-2	-2	0.25	0.88
Organization Scalability	2	1	2	0	2	1	-1	0	-2	-1	-2	1	0.25	0.69
Reduce Truck Factor	2	1	1	-1	1	2	1	1	-2	1	-2	-2	0.25	0.65

4.2 RQ2: *What practices fall within the scope of DevOps?*

One of the goals with the literature review has been to explore DevOps practices in order to answer **RQ2**. Two main papers were proposed to derive the practices from: the systematic mapping study by Jabbari et al. [13, p. 6] and the Systematic Literature Review by Erich et al. [5, p. 15]. Jabbari et al. has suggested some DevOps practices through a Systematic Mapping Study [13, p. 6]. Through their definition of DevOps, they have found that communication and collaboration as well as automation are some of the central parts [13, p. 6]. The study by Erich et al. suggests that culture and automation are the most central parts when adopting DevOps [5, p. 15], based on six interviews in different organizations. In their SLR, Erich et al. state that *"there is no DevOps process or methodology"* [7, p. 16] and *"DevOps is not a one size fits all solution to solve a problem in software engineering like Scrum is for example."* [7, p. 16]. Due to this, it is hard to give a specific answer to **RQ2** based on literature, although there seems to be some areas that are more common than others. Damon Edwards and John Willis coined the acronym CAMS (Culture, Automation, Measurement, and Sharing) after the first Devopsdays conference [32], but in Erich et al.'s interviews, *"measurement, monitoring, and sharing were [only] considered important by most but not all interviewees."* [5, p. 15].

To sum up, there are some DevOps practices defined in literature [13, p. 6][7, pp. 17-18]. In these studies, areas like culture of collaboration between Dev and Ops, and automation, seem to be central concepts.

With the results from the literature review in mind, one of the goals of the interviews was to find out what practitioners have done when implementing DevOps in order to derive a set of practices. Several practices emerged from thematically coding the transcriptions. The practices were then grouped in various overarching categories. An in-depth description of each practice can be seen in Table 4.4. Table 4.5 shows what practitioners affirmed the different practices during the interviews.

Table 4.4: This table describes the practices that emerged from the interviews, with a few quotes to exemplify the underlying data. The quotes have been translated from Swedish to English.

Practice	**Description**
	Architecture
Infrastructure as Code	Having the infrastructure declarative in source files, such that the setup of the infrastructure is repeatable. *" ...with the possibilities that exists today, especially with the Cloud, the server is just a few lines of code. So, we automatize these tasks that were done manually in the past. You get the infrastructure as code. "* — Practitioner J *"You cannot test a bullet point list since you cannot ensure it is run the same way every time, but if it is in a system that just executes these things then you can test it. [...] It gets continually verified. Add a requirement, put it in the script and it will be tested so we can detect errors immediately. [...] You remove administrative noise. [...] Everything gets focused on the application and less on the infrastructure, which means, from my perspective, that it is easier since there are fewer moving parts that I need to know about."* — Practitioner K
Loosely-coupled Architecture	Organize the architecture to avoid monoliths, using for example a microservices design. *"If you build a monolith, it gets harder to release, so typically when looking at DevOps adoption and one looks into how to organize teams, one also looks at how the application can be divided."* — Practitioner B
	Automation
Continuous Delivery	Working towards continually deliver deployment ready code. Working towards all code being releasable at any point.
Automatic Testing	Working towards automated testing of all code.
Automatic Deployment	Automating the deployment of releasable code into the production environment.
Continuous Integration	Continually integrate code branches into the master branch.
Continuous Deployment	Working towards continually deploy into production.
	Culture & Organization

...

Practice	Description
Fail-fast	Working towards reducing fear of making mistakes and instead embrace learning from mistakes. Deploy, and then fix the detected errors swiftly by deploying a new fix. This contrasts with the thinking where failures late in the cycle are expensive and must be avoided at all cost.
Cross-Functional Teams	Working towards having all the knowledge needed for both Dev and Ops tasks in the same team. The team takes responsibility from requirements, to deployment to maintenance and support. Sometimes this is further broadened to include business analysts (BAs), Quality Assurance (QA), etc. *"I think that in the cleanest form, DevOps is empowered teams. In other words that the teams have the responsibility to work with the entirety of a solution. From conception, till you build it, till you also deliver it and take responsibility for operations. From start to finish. With everything that entails."* — Practitioner B *"You get more focus on what we really are doing. Are we developing software, operating systems, or simply generating business value? If you look at it today, we got teams with developers, operations, designers, UX, business analysts, economics and with GDPR lawyers in the team and suddenly, everyone is there. [...] We cannot segment ourselves into different departments, we must work together and with so much more than just technical aspects."* — Practitioner K
Broaden the Developer Role	Working towards Dev embracing tasks that traditionally belongs to other roles. Areas normally include making Dev more responsible for testing, deployment and support. Working towards having all the knowledge needed for both Dev and Ops tasks in the same team. The team takes responsibility from requirements, to deployment to maintenance and support. Sometimes this is further broadened to include business analysts (BAs), Quality Assurance (QA), etc. *"I would say Dev are taking over the Ops aspects. [...] Normally in large companies it is a gradual process, and some old systems may still be operated as before. But it is less and less of that [...] causing a natural flow of Ops [tasks] into DevOps teams so there is a risk that experts on operating old systems will not be needed long term. [...] I discussed the other day at a workplace where I accidentally said that when you work DevOps there are constellations without testers. The test lead then immediately spoke up and said, 'shall we not test our products anymore?'. Yeah that was not exactly what I meant. I mean the tester needs a testing role when he is developing. If we want shorter lead-times we cannot code, code wrong, and then wait for someone to eventually test it."* — Practitioner B
Goal Sharing	Dev and Ops more explicitly share the same goal of meeting the need of the customers. This could be through a collaboratively defined definition of done between Dev and Ops, a cross-functional team that together takes responsibility for Dev and Ops tasks and so on.
Knowledge Sharing	Working towards sharing knowledge and learning from each other. This can be done by establishing communities of practice, by having Dev and Ops coach each other and so on.

...

Practice	Description
Shared Responsibility	Working towards Dev and Ops having shared responsibilities. This can look like Dev taking responsibility for what they deliver as it is deployed and running. For example, when the software crashes Dev gets involved to solve and learn instead of brushing it of as Ops responsibility.
Pioneer DevOps Team	Start adopting DevOps by creating a "shining example" team that other teams can resemble.
	Monitoring & Data-driven Development
System Monitoring	Monitor how the system is doing (performance, down-time, ...). This can be done on different levels (infrastructure, features, ...). *"Monitoring and monitoring are for me actually two parts. It's the system in itself, but also users, systems... I want it to monitor how the system as such, perhaps with dummy users or similar, how does the system work as such if we have no load at all but just put on our own stuff. [...] Monitoring from a system perspective that it works; it's what Ops want."* — Practitioner G
User behavior Monitoring	Work towards monitoring how the users are using the system to provide feedback to Dev. *"But monitor our users to learn something about how they use the system. That's also good. The user perspective is what Dev wants."* — Practitioner G
Hypothesis-driven Development	Let the development be oriented on the experimentation of hypotheses. Set up a hypothesis, test it on all or some selected users and observe the results. *"It is really when testing a hypothesis. So, if you don't know anything, it is better to do as little as possible to test something and measure something and see 'Is this something worth going for?'."* — Practitioner A
	Other
Versioning of all Software Artifacts	Working towards version controlling everything: source code, configuration files, infrastructure, dependencies, etc.
Adoption backed up by leadership	Leadership actively supports the process of adopting DevOps.
Clear DevOps Vision	Leadership has established and clearly communicates the vision of going towards DevOps.
Feature Toggling	Also known as feature flags. Enables turning on and of features in runtime avoiding long-lived feature branches, and the possibility of disabling problematic features. Features can be tested and deployed before they are finished.

Table 4.5: The table shows DevOps practices that have been identified in the interview data. Every practice affirmed by the practitioner is marked with (×). Each row represents an identified DevOps practice, and each column represents an interviewed practitioner (found in Table 3.2). The practices are grouped by overarching categories.

Category	Practice	A	B	C	D	E	F	G	H	I	J	K	L
Architecture	Infrastructure as Code	×	×	×		×		×	×	×	×	×	×
	Loosely-coupled Architecture	×	×	×						×	×		×
Automation	Continuous Delivery	×	×	×		×		×		×	×	×	×
	Automatic Testing	×	×	×	×	×	×	×			×	×	×
	Automatic Deployment		×	×	×			×				×	
	Continuous Integration			×		×	×	×		×			×
	Continuous Deployment				×	×	×	×		×		×	×
Culture & Organization	Fail-fast							×	×		×		
	Cross-Functional Teams	×	×			×		×	×	×		×	×
	Broaden the Developer Role	×	×	×		×	×	×	×	×	×	×	×
	Goal Sharing	×			×	×		×	×	×		×	
	Knowledge Sharing	×			×	×		×	×	×		×	×
	Shared Responsibility	×		×	×	×	×	×	×	×	×	×	×
	Pioneer DevOps Team					×	×		×	×	×		
Monitoring & Data-driven Development	System Monitoring	×	×	×	×	×	×	×	×	×	×	×	×
	User behavior Monitoring	×	×	×	×	×		×	×	×		×	
	Hypothesis-driven Development	×	×	×	×			×		×			
Other	Versioning of all Software Artifacts	×		×				×	×	×	×		×
	Adoption backed up by leadership	×	×	×		×			×	×			×
	Clear DevOps Vision					×				×			
	Feature Toggling		×			×						×	×

4.3 RQ3: *What is the criticality of DevOps practices for the adoption goals?*

To understand what DevOps practices are more critical, the practices that emerged from the interviews, as found in Table 4.5, were used as input to the survey. Table 4.6 shows how important the respondents have considered each practice for them to reach their DevOps adoption goal(s). This was done to understand the criticality of the practices for reaching the adoption goals. What can be noted in the table is that there is emphasis on automation tasks such as Continuous Delivery, Automatic Testing and Automatic Deployment as well as System Monitoring. On the other hand, User behavior Monitoring and Hypothesis-driven Development are not considered as central with the average scores of 0.27 respectively 0.25.

Table 4.6: The shows how important each practice has been for each practitioner **A**–**L** (found in Table 3.2) in order for them to reach their goals found in Table 4.1. Each practice is ranked by importance by practitioners using an ordinal scale from strongly agree to strongly disagree, here represented as numbers ranging from 2 to −2. A hyphen indicates "No opinion". The reason that no data is available for Feature Toggling is that the question accidentally was omitted during the questionnaire construction phase. The practitioners are grouped by DevOps experience on the ordinal scale from very high to mid (as can be found in Table 3.2). The W. Avg is the weighted average weighed by experience (Very High: 3, High: 2, Mid: 1).

	DevOps Experience	Very High (3)						High (2)		Mid (1)				Avg	W. Avg
Category	Practice	A	B	C	I	J	L	E	K	D	F	G	H		
Architecture	Infrastructure as Code	2	2	2	2	2	2	-1	-1	0	0	2	1	1,08	1,35
	Loosely-coupled Architecture	2	1	2	1	2	2	0	0	1	1	0	1	1,08	1,27
Automation	Continuous Delivery	2	2	2	2	2	2	2	1	2	2	2	-	1,91	1,92
	Automatic Testing	2	1	2	2	2	2	2	1	2	1	2	2	1,45	1,77
	Automatic Deployment	2	2	2	2	2	2	1	1	1	2	2	2	1,75	1,81
	Continuous Integration	2	2	2	2	2	-2	2	0	2	2	2	-	1,75	1,36
	Continuous Deployment	2	1	-1	1	2	1	1	-1	2	2	2	-	1,09	0,96
Culture & Organization	Fail-fast	2	2	-1	1	2	1	1	2	0	2	1	1	1,17	1,19
	Cross-Functional Teams	2	2	1	0	2	2	2	1	-2	2	0	2	1,17	1,35
	Broaden the Developer Role	2	2	1	1	2	2	1	1	-2	2	1	1	1,17	1,38
	Goal Sharing	2	2	0	2	1	1	0	1	0	2	1	1	1,08	1,15
	Knowledge Sharing	2	1	0	1	2	0	1	2	0	2	0	1	1,00	1,04
	Shared Responsibility	2	2	1	0	1	1	0	2	0	1	0	2	1,00	1,08
	Pioneer DevOps Team	1	2	-1	-1	2	1	0	0	-2	2	0	1	0,42	0,50
Monitoring & Data-driven Development	System Monitoring	2	2	1	2	2	2	2	1	0	1	2	-	1,55	1,68
	User behavior Monitoring	2	1	-1	-	0	0	1	-1	-2	0	2	1	0,25	0,30
	Hypothesis-driven Development	0	0	0	1	0	0	-1	0	0	1	0	2	0,27	0,15
Other	Versioning of all Software Artifacts	2	2	2	2	0	1	0	0	0	1	2	-	1,09	1,20
	Adoption backed up by leadership	2	2	1	2	2	2	0	0	-1	1	-2	1	0,83	1,23
	Clear DevOps Vision	2	2	1	2	1	2	-2	-1	0	0	-2	0	0,42	0,85
	Feature Toggling	Data Not Available													

When looking at how the level of experience among the practitioners affects the results, the variance between the groups is not that high. Worth to notice is that the practitioners with high level of experience in adopting DevOps has a notably higher average on Infrastructure as Code and Loosely-coupled Architecture, as is indicated by the weighted average. The same thing goes for Adoption backed up by leadership

and Clear DevOps Vision. An outlier regarding Continuous Integration makes the weighted average go down a bit. User behavior Monitoring and Hypothesis-driven Development stays on the lowest values even when accounting for level of experience.

4.4 RQ4: *What is the perceived impact of implementing DevOps practices?*

When it comes to the perceived impact of DevOps, there are various types of data that can be presented. Firstly, there are self-reported effects from the survey by the practitioners. This can be connected to the goals making way for a causal relationship. Next, there is the elicited effects from the interviews that are more general in nature.

In Table 4.7, the results from the survey is presented regarding the effects the practitioners have experienced from adopting DevOps. More than just the average, a weighted average is presented to put more weight into what the more experienced practitioners have experienced. Looking at the average, many agree the they have experienced a positive change in Throughput, Product Quality and Work Environment. Moreover, though most answers are positive, there are some people who disagree on observing a change in Product Scalability, Organization Scalability, and Reduce Truck Factor. Comparing the average to the weighted average does not make that much difference, since there were little data reported by the less experienced practitioners.

Table 4.7: The table indicates what effects the practitioners have experienced when adopting DevOps in relation to the different goals. The impact of each of the goals have been ranked by the practitioners using an ordinal scale from strongly agree to strongly disagree, here represented as numbers ranging from 2 to −2. A hyphen indicates "I don't know" and the ∅ symbol means that no answer has been provided by the respondent. The practitioners are grouped by experience on the ordinal scale from very high to mid (as can be found in Table 3.2). The W. Avg is the weighted average weighed by experience (Very High: 3, High: 2, Mid: 1).

DevOps Experience	Very High (3)						High (2)		Mid (1)				Avg	W. Avg
Goal	A	B	C	I	J	L	E	K	D	F	G	H		
Throughput	2	2	1	1	1	1	2	1	∅	1	2	0	1.27	1.32
Product Quality	2	1	1	1	0	2	2	0	∅	-	-	1	1.11	1.13
Value of Delivery	2	2	0	0	0	1	1	0	∅	-	-	1	0.78	0.78
Work Environment	2	1	1	2	0	2	2	0	∅	1	2	1	1.27	1.28
Product Scalability	2	-	2	0	0	1	0	-1	∅	-	0	∅	0.50	0.65
Organization Scalability	2	∅	2	1	0	1	0	1	∅	-	-	-1	0.75	0.95
Reduce Truck Factor	2	∅	0	0	0	1	1	-1	∅	1	1	1	0.60	0.55

While there is no data to clearly connect practices to specific effects, one can suggest the existence of a cause-effect relationship between practices perceived as critical and the effects experienced by the practitioners. This suggested relationship is illustrated in Figure 4.1. The validation for this relationship is left for further research.

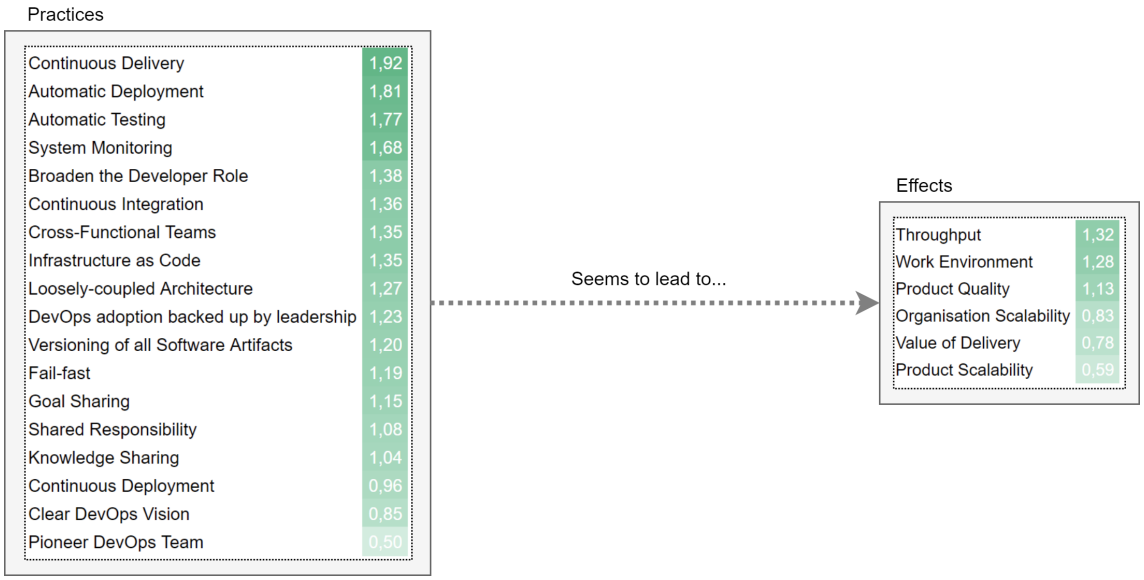

Figure 4.1: A proposed cause-effect relationship between practices and effects. Both practices and effects are ranked from 2 to −2, strongly agree to strongly disagree. The practices are ranked by their criticality for adoption goals according to the practitioners, and the effects are ranked according to the perceived effects according to the practitioners, both weighted by their experience.

Apart from the results reported in the survey, various general effects were elicited during the interviews. In Table 4.8, effects mentioned in the interviews are reported. The effects reported are based on what the interviewed practitioners have perceived while adopting DevOps.

Table 4.8: The table shows a list of positive and negative effects extracted from the interview data. The effects are grouped into different areas of impact, and the positive things are marked with a + and the negative with a −.

Area of impact	**Effects**
Throughput	+ Shorter lead times. + Increased transparency: Inefficiency in the value stream is highlighted. − Development speed might be lower at the beginning before getting used to the way of working.
Product Quality	+ Higher quality of the product. + Fewer and shorter downtimes. + More focus on testing while coding. Solve bugs instead of patching them.
Value of Delivery	+ Gives a greater understanding of the whole value stream. + Increased sense of ownership. + More feedback from the system and the users.

...

Area of impact	Effects
Work Environment	+ Higher well-being. + Greater sense of responsibility. + Automation of repetitive tasks. + Reduced deployment pain. + Funnier way of working according to some. + No "blame game": When something fails, developers cannot blame operations and vice versa. Failing gives a learning opportunity. + Closer collaboration by working together. − Developers might need to be in support, either 24/7 or during office hours. Having to be signed up during night time now and then is not very popular. − Operations personnel in cross-functional teams might feel superfluous when just waiting for developers to ask questions. − Creates a sense of inadequacy among some. − If someone gets too much to learn, they might feel that it is overwhelming. − People become generalists and might not feel as important anymore. − Not a fun way of working according to some. − Some mean it is stressful to always focus on the bigger picture and be involved at all times.
Product Scalability	+ One can hire more developers without hiring more operations personnel if developers are doing the deployment tasks.
Organization Scalability	+ One can hire more developers without hiring more operations personnel if developers are doing the deployment tasks.
Reduce Truck Factor	+ More automation means less external documentation since deployment knowledge can be scripted. + Peace of mind: You are more exchangeable. If everything is scripted in code, specialists can be sick or on vacation without risking being disturbed.
Other	− It takes time and costs to learn delivery pipeline tools. − Delivery pipeline tools cost money. − It can be hard to find people skilled with delivery pipeline tools. − Initially, it takes the time to set up a delivery pipeline. − Financing models need to be changed when not working according to a waterfall model. − Over time, operations personnel may lose their job when developers take over the tasks.

4.5 RQ5: *To what extent can the impact of adopting DevOps practices be measured?*

What would give more weight to the perceived effects in Section 4.4 would be measurements confirming them. During the interviews, some practitioners stated that they or someone else around them had measured some things that can be seen in Table 4.9. Worth to note is that most of the metrics that have been used are related to Throughput. The only other used metric is related to Product Quality. On the other hand, when asking the practitioners, both in the interviews and in the survey, for suggestions on how the impact can be measured, some more metrics came up as shown in the table.

Table 4.9: The table shows used and suggested metrics of how the effects on different adoption goals have been/can be measured. The data of the used metrics comes from the interviews, and the data of the suggested metrics comes from the interviews and from comments in the survey.

Goal	**Used metrics**	**Suggested metrics**
Throughput	• time from requirement to deploy • how often code is checked in • how often there are release candidates • time between deploys • number of deploys per time unit • time between code commit and deploy, and build times	• number of tasks a cross-functional team can perform without having to ask for help and wait for an external person • degree to which the delivery tool chain is automated • mean time until delivery • number of keystrokes to deploy — goal is one click deploy • time from idea to working software in production • time to do a rollback (when things go wrong)
Product Quality	• defects in production over a certain time period	• mean time to detection of an error • number of bugs during development • number of defects after release
Value of Delivery		• time from deploy until there is a knowledge if it was useful
Work Environment		• overtime due to deployment work
Product Scalability		
Organization Scalability		
Reduce Truck Factor		• number of tasks a cross-functional team can perform without having to ask for help and wait for an external person • release frequency and by who

Although there are some relevant metrics, they are not widely used. Nevertheless, some common reasons for using them was for the practitioners to convince their clients to understand their need for e.g. better Throughput, or to see trends if they were doing progress. Other than that, some practitioners suggested to look at the metrics from the State of DevOps Report 2016, which are described further in Chapter 2. To sum up, there are some ways to measure the impact of DevOps adoption, but the metrics are not used that much.

5

Discussion

In this chapter, there will be a discussion regarding the results of the study, going into goals, DevOps practices, and the effects of adopting these. This will be followed by a section regarding threats to validity, and at last some suggestions for future research will be provided.

5.1 Adoption Goals Broader Than Anticipated

Looking at the literature the major goals with DevOps adoption seemed to be Throughput with maintained Product Quality [5, p. 1]. However, while conducting the interviews several more goals emerged as can be seen in Table 4.2. They were further verified in the survey as seen in Table 4.3. While Throughput and Product Quality remains the most important goals, practitioners also widely consider the Value of Delivery and Work Environment as important.

Another interesting finding is that the adoption goals seem to vary depending on the practitioners' experience with DevOps. Considering the experience level, Product & Organization Scalability as well as Reducing Truck Factor increases in importance. This is likely partly due to that the practitioners with less experience have yet to see any effect in these areas as seen in Table 4.7.

It is encouraging to see that DevOps adoption benefits might go beyond what was initially expected based on existing literature. What especially stood out was that Work Environment is a big factor for adoption.

5.2 DevOps Practices

One of the goals of the thesis was to define a set of DevOps practices and begin to understand their criticality for the adoption goals. These practices can be found in Table 4.6. There are some findings here that are especially interesting.

5.2.1 Automation is Central

As can be seen in Table 4.6, a central category of practices is automation, centered around automating all aspects of the delivery pipeline. Most of the practitioners strongly agree that practices like Continuous Delivery, Automatic Testing, Automatic Deployment and Continuous Integration have been important aspects in helping them to reach their goals. The outlier (L) regarding Continuous Integration is probably due to a misunderstanding of the practice. This is supported by a comment the practitioner made in the survey, that they are developing software trunk-based without any feature branches. Thus, the practitioner could strongly disagree to our explanation of Continuous Integration being *"Continually integrate code branches into the master branch"*.

5.2.2 Cultural and Organizational Practices Transforms Ops

While the literature review led to an understanding that cultural and organizational change towards improved collaboration are considered key, we found that it almost always involves significant organizational change with regards to the Ops tasks. In almost all cases, the practices of either Broaden the Developer Role or establishing Cross-Functional Teams, or both, are considered central as can be seen in Table 4.6. With emerging tooling and technologies, the Ops tasks are being transformed. In many companies, traditional Ops have been completely replaced with only developers taking up the responsibilities of Ops. This is often combined with outsourcing of the infrastructure to cloud services. While DevOps is argued in the literature to be about bridging the gap between Dev and Ops [13, p. 5], it would rather seem like Ops is being transformed into something different that fits within the Dev silo. In some cases, according to the results, this involves moving experienced Ops into cross-functional teams, but in other cases Dev personnel simply absorbed the Ops responsibilities. This comes with a various set of effects, where sometimes Dev underestimates the amount of work there is with operating software, while increasing their product quality, work environment and rate of delivery.

Large organizational change in Ops are not always the case however. There are examples of organizations still working with Dev and Ops as separate departments that we found. Then focus lies on Knowledge Sharing and Goal Sharing, where even if you are in development you are still responsible for your work after it has been handed to Ops. Sometimes teams get geographically co-located to increase collaboration and in some cases one or two Ops persons are placed in each Dev team. The purpose is to improve Knowledge Sharing so that Dev understands what deliverable code means while they can also meet the needs of Ops with regards to logging and similar.

We found it quite fascinating that there seem to be a general shift towards developers not only embracing the testing role as has been seen previously in agile concepts, but now also the operation tasks. It is probable, when looking at our data, that operation departments as they exist today may not be as common in the future. Consequently,

it is worth to consider the ethical implications of this change. There is an inherent risk that as the traditional operations role becomes deprecated, people will lose their jobs unless they adapt. This dilemma was mentioned during the interviews and is listed as a negative effect in Table 4.8. This study might accelerate this change by bringing this to light. However, as the industry is ever evolving, it is expected that disruptive changes will continue to occur. This change should all in all be beneficial enabling faster delivery of better software.

5.2.3 Data-driven Development is Attractive but Immature

In order to increase Value of Delivery, participants talk about making use of feedback from systems and users. While System Monitoring is a widely applied practice confirming the findings by Erich et al. [5, p. 15], practices related to making use of user data is not. Examples of such practices are User Behavior Monitoring and Hypothesis-driven Development. Even though these practices were positively affirmed during the interviews by several practitioners as seen in Table 4.5, they do not seem to be critical when looking at the survey results in Table 4.6. One explanation for this could be, in some cases, the immaturity of the software the practitioners have been working on. There might not be enough production data that can be gathered. Another reason could be that the practitioners' clients are not ready for taking the step yet towards Data-driven Development. It could be that the delivery rate is still too low for it to be feasible, or that there are policies, physical constraints (e.g. production environment has no Internet connection), outsourced Ops, cross-company barriers, etc. preventing user data information from being gathered. Lastly, it could be that the practitioner does not have enough experience in performing the practice but wants to. Given these points, research could investigate this area further looking for impediments and/or validate the effects of applying such practices.

5.2.4 Architecture Affects Adoption

Recurring in the interviews was the question of architecture and its effect on delivery. Often large code monoliths seem to hinder fast delivery of software. Hence, practitioners to a great degree attempts to make the architecture loosely-coupled as can be seen in Table 4.6.

In literature, DevOps is often associated with a service-based architecture [5, p. 15]. Erich et al. state that they have found that such an architecture is rarely considered as a core part of DevOps, but is still helpful [5, p. 15]. The data gathered in this study confirms this finding. While a service-based architecture is not considered a core part of DevOps according to the practitioners, what is considered a core part is to make the architecture less coupled, where services is one of the ways it can be done.

5.2.5 Experienced Adopters Values Leadership Vision and Support

Looking at Table 4.6, it seems like the practitioners have had different experiences when it comes to having a Clear DevOps Vision. Only the practitioners with a high level of experience in adopting DevOps agree or strongly agree that this has been an important practice for them to reach their goals.

Furthermore, looking at Adoption backed up by leadership, the data looks relatively similar to Clear DevOps Vision. Though, some of the low values here could be due to the practitioners not having experienced support from leadership in their adoptions. That means that they still might value it. However, for the practitioners with a high experience level, everyone reports that it has helped them in reaching their goals. Putting this in relation to the State of DevOps Report, it confirms their finding regarding the importance of leadership [11, pp. 12-19].

5.3 Effects of Adopting DevOps Practices

Looking at the data for the effects of adoption perceived by the practitioners, all but two agreed or strongly agreed that they experienced improved Throughput as seen in Table 4.7. All other goals scored high as well. What stands out is the improved Work Environment that ranks second highest, almost as high as Throughput, while only being in fourth place for goals considered when adopting DevOps. Work Environment seems to be one of the key aspects worth looking at with regards to DevOps adoption.

It is further worth noting that many of the less experienced adopters have not reported any effects by either skipping the question or answering, “I don’t know”. This is likely due to their adoption contexts being immature or that they have yet to start to measure and/or think about the effects.

Looking at the criticality of the practices for their adoption goals and the perceived effects, we propose the cause-effect relationship presented in Figure 4.1. Even though the proposal is based on qualitative data with no explicit interconnections between the practices and the effects, this can be a good resource for further research in for example setting up controlled experiments involving the practices.

Another interesting note is that it would seem that few practitioners measure the effects of adopting DevOps practices. While some metrics are proposed in Table 4.9, they are not broadly adopted if at all. There are various reasons for this, but mainly it is tedious, and questionable if it is helpful according to some practitioners. It is argued that metrics often cause sub-optimization and drive unwanted behavior. For example, maximizing the number of deploys per time unit does not provide value if it is not the right thing that is delivered. In the same way, it is unclear whether one wants to find many or few defects in production over a certain amount of time. While it is good to find them to be able to fix them, one does not want them to be

there from the very beginning. Despite the importance of measuring practices being disputed by some practitioners, it is useful for evaluating the criticality of practices. Here, one focus in future research could be to elicit more measurements for Value of Delivery and Work Environment considering they are high ranking adoption goals.

From the interviews, both positive and negative aspects were brought to light, as can be seen in Table 4.8. The way negative aspects like overhead work, lack of engagement among some people, developers in support, change of financial models, etc. affect the organization performance could be an area for future research. What can be noted is that there are no reports of negative aspects when it comes to the areas Product Quality, Value of Delivery, Product Scalability, Organization Scalability nor Reduce Truck Factor. This may be due to a positive bias from the practitioners' side towards DevOps adoption, lack of enough data, or there are no negative aspects worth reporting regarding these. As expected, based on the results from Table 4.7, there are mostly positive aspects related to Throughput. Regarding Work Environment, on the other hand, there are both many positive and many negative effects reported. Some people seem to like the possibility of having more responsibility, while some do not see the value of the change, or that it is too much new things to learn. Finally, looking at other effects that cannot be categorized anywhere else, there are mostly things related to expenses and financial things.

5.4 Threats to Validity

In the following section, threats to the validity in the study will be addressed. The analysis of different kinds of threats dealt with is inspired by Lewis [20], who is taking inspiration from Maxwell [22], and Runeson et al. [28].

General Considerations

Some threats to validity lies with the sampling of the cases. The methodology assumes that the practitioners sampled are knowledgeable in the field of DevOps and representative of DevOps practitioners. The varied experience level among the sampled practitioners posed as such a potential issue. There was an attempt to remedy this by weighting the results based on experience level. Another threat is that all sampled practitioners were selected from a narrow geographical area, as such this may affect the generalizability of the findings.

Another threat is that the practitioners were asked to answer the questions based on how it relates to their previous experience from work they have performed in the past. For example, when asking about the criticality of a practice, it was asked in relation to previous adoptions. Due to the field of Software Engineering moving fast, this information may already be old news. The upside is that the answers are more connected to the actual perceived outcomes.

Descriptive Validity

To make sure all information from the interviews was understood correctly, they were all recorded. To make sure no data was lost due to bad quality, two recorders were used. All interviews were held in quiet rooms, with no one else present than the thesis writers and the practitioners. In some interviews, the practitioner used a white board to describe their thoughts. These were photographed and taken into consideration during the analysis of the data.

Interpretation Validity

When creating the interview guide, the questions were made as open as possible to let the interviewees explain their experiences of DevOps adoption. However, during the interviews, some suggestions were given of different categories of practices to help inspire the interviewees. These were first presented after the interviewee had presented their own thoughts (except in the first interview, where by mistake these were visible in the interview guide so that the interviewee could see them). These questions were used both to inspire the practitioners through exemplification, but also to gain an understanding of practices mentioned by other practitioners, as well as validate the findings in the literature review.

Theory Validity

Due to DevOps being a popular buzzword in the industry, there is a chance that the thesis writers lean towards a bias thinking positively about DevOps. However, by being students with no prior experience or deep knowledge of the topic, there are no ambitions in supporting one theory or another.

Next, due to questions in the survey being written in a positive manner, they could be misleading. Answering Neutral, Disagree or Strongly disagree to, for example, the question *"To reach my DevOps adoption goal(s), I have considered the following practices important"* might mean different things. Disagreeing could mean that the practice has negatively impacted the goal or definitely not impacted the goal. That might have affected making the practices getting a lower score since we assume disagreeing means a negative impact. This was an unfortunate mistake done when constructing the survey.

Researcher Bias, Reactivity and Reliability

Due to little prior knowledge on the subject, we brought little bias coming into the thesis. There might be a bias from the practitioners due to them wanting to please us as students, partly because they could have seen the interviews as a recruitment opportunity.

Internal Validity

Looking at the cause effect relationship proposed in Figure 4.1, there is nothing in the data pointing out what practices gives what effects. For example, even though all practitioners consider Automatic Testing as important and all practitioners see an increase in Throughput, it might be due to another practice also implemented. All we can say is that the practices together, along with other potential factors, seem to lead to all the goals together.

External Validity

When constructing the thesis there was little focus on the size of the organizations adopting DevOps. While the sampled practitioners have been involved in a wide range of companies of different sizes, the thesis results might suffer from less generalizability due to not accounting for organizational size.

5.5 Future Work

Future work could focus on setting up experimental studies or similar considering the proposed cause-effect relationship presented in Figure 4.1. Similarly, one could use some of the derived metrics in Table 4.9 in order to gather quantitative data for the effects of DevOps practices. Other areas of research that could be of interest based on the findings of this thesis would be how Ops is being transformed, how DevOps practices affect work environment and effects and/or impediments of different Data-driven Development practices. While some negative aspects are presented in this thesis, one could investigate the negative aspects of adopting DevOps practices in more depth.

6

Conclusion

The purpose of the study has been to evaluate goals of adoption and suggest practices to be considered for implementation in order to reach the goals.

We have found that while there is not a clear definition of why DevOps is adopted, there is a set of goals practitioners tend to agree upon as seen in Table 4.3:

- Throughput and Product Quality are the top goals for adoption, as expected with regards to existing research.
- Value of Delivery and Work Environment are considered important by all practitioners, followed by Organizational & Product Scalability and Reduce Truck Factor for the experienced practitioners.

We have found that the practices performed varies between the practitioners, where some have more consensus in their application than others. We have identified key areas unilaterally discussed when implementing DevOps, with corresponding practices as seen in Table 4.6:

- **Architecture:** Considered by all experienced practitioners, where avoiding monoliths and Infrastructure as Code is in focus.
- **Automation:** Automation of the delivery pipeline is central to all practitioners, which validates existing research.
- **Culture & Organization:** Cultural practices for increased collaboration such as Goal Sharing are central, which validates existing research. In addition, practices affecting the organizational structure, specifically Cross-Functional Teams and Broaden the Developer Role, are considered important by practitioners having a big effect on the nature of Ops tasks.
- **Monitoring & Data Driven Development:** Is considered, which validates existing research for monitoring practices. In addition, practitioners talk positively of Data-driven Development, but the industry does not seem ready in all places yet.
- **Other**: Leadership is important to experienced practitioners, such as support from leadership and a clear vision. Versioning of all artifacts is considered important by the practitioners.

We have found that the effects of implementing DevOps practices are rarely measured and few metrics are used, especially by the less experienced practitioners. However, all but one practitioner reports having perceived some impact on the adoption goals as seen in Table 4.7. The main effect is on Throughput and Work Environment, but practitioners have also perceived a positive effect on all other adoption goals. These findings could be important for organizations' decision making on adopting DevOps.

We have established a potential cause-effect relationship between the practices and goals as seen in Figure 4.1 that can be useful in future research.

Bibliography

[1] John W Creswell. *Research design: Qualitative & quantitative approaches*. Thousand Oaks, CA, US: Sage Publications, Inc, 1994, pp. xix, 228–xix, 228. ISBN: 0-8039-5254-6 (Hardcover); 0-8039-5255-4 (Paperback).

[2] Patrick Debois et al. "Devops: A Software Revolution in the Making?" In: *Cutter IT Journal* 24.8 (2011), pp. 1–40. ISSN: 15227383. URL: https://www.cutter.com/sites/default/files/itjournal/fulltext/2011/08/itj1108.pdf.

[3] Devopsdays. *devopsdays*. 2018. URL: https://www.devopsdays.org/ (visited on 05/16/2018).

[4] Steve Easterbrook et al. "Selecting Empirical Methods for Software Engineering Research". In: *Guide to Advanced Empirical Software Engineering*. London: Springer London, 2008, pp. 285–311. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5_11. arXiv: arXiv:1011.1669v3. URL: https://link.springer.com/chapter/10.1007%7B%5C%%7D2F978-1-84800-044-5_11.

[5] F. M.A. Erich, C. Amrit, and M. Daneva. "A qualitative study of DevOps usage in practice". In: *Journal of Software: Evolution and Process*. Vol. 29. 6. John Wiley & Sons, Ltd., 2017, pp. 1–20. ISBN: 9781450330565. DOI: 10.1002/smr.1885. arXiv: 1408.1293. URL: http://doi.wiley.com/10.1002/smr.1885.

[6] Floris Erich, Chintan Amrit, and Maya Daneva. "Cooperation Between Software Development and Operations: A Literature Review". In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. Torino, 2014. URL: https://www.researchgate.net/publication/280712228_20140526_Cooperation_Between_Software_Development_and_Operations_A_Literature_Review.

[7] Floris Erich, Chintan Amrit, and Maya Daneva. *Report: DevOps Literature Review*. Tech. rep. University of Twente, 2014, pp. 1–27. DOI: 10.13140/2.1.5125.1201. URL: https://www.researchgate.net/publication/267330992_Report_DevOps_Literature_Review.

[8] Nicole Forsgren Velasquez et al. *2014 State of DevOps Report*. 2014. URL: https://puppet.com/resources/whitepaper/2014-state-devops-report.

[9] Nicole Forsgren and Jez Humble. "DevOps: Profiles in ITSM Performance and Contributing Factors". In: *SSRN Electronic Journal* (Oct. 2015). ISSN: 1556-

5068. DOI: 10.2139/ssrn.2681906. URL: http://www.ssrn.com/abstract=2681906.

[10] Nicole Forsgren et al. *2016 State of DevOps Report*. 2016. URL: https://puppet.com/resources/whitepaper/2016-state-of-devops-report.

[11] Nicole Forsgren et al. *2017 State of DevOps Report*. 2017. URL: https://puppet.com/resources/whitepaper/2017-state-of-devops-report.

[12] Harvard University. *Strategies for Qualitative Interviews*. URL: http://sociology.fas.harvard.edu/files/sociology/files/interview_strategies.pdf (visited on 03/09/2018).

[13] Ramtin Jabbari et al. "What is DevOps?" In: *Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops*. New York, USA: ACM Press, 2016, pp. 1–11. ISBN: 9781450341349. DOI: 10.1145/2962695.2962707. URL: http://dl.acm.org/citation.cfm?doid=2962695.2962707.

[14] Gene Kim. *Top 11 Things You Need To Know About DevOps*. 2013. URL: http://www.bogotobogo.com/DevOps/Puppet/images/DevOps/Top11ThingsToKnowAboutDevOps.pdf.

[15] Barbara A. Kitchenham and Shari Lawrence Pfleeger. "Principles of Survey Research Part 2: Designing a Survey". In: *ACM SIGSOFT Software Engineering Notes* 27.1 (Jan. 2002), pp. 18–20. ISSN: 01635948. DOI: 10.1145/566493.566495. URL: http://dl.acm.org/citation.cfm?doid=566493.566495.

[16] Barbara A. Kitchenham and Shari Lawrence Pfleeger. "Principles of Survey Research Part 3: Constructing a Survey Instrument". In: *ACM SIGSOFT Software Engineering Notes* 27.2 (Mar. 2002), p. 20. ISSN: 01635948. DOI: 10.1145/511152.511155. URL: http://portal.acm.org/citation.cfm?doid=511152.511155.

[17] Barbara Kitchenham and Shari Lawrence Pfleeger. "Principles of Survey Research Part 4: Questionnaire Evaluation". In: *ACM SIGSOFT Software Engineering Notes* 27.3 (May 2002), p. 20. ISSN: 01635948. DOI: 10.1145/638574.638580. URL: http://portal.acm.org/citation.cfm?doid=638574.638580.

[18] Barbara Kitchenham and Shari Lawrence Pfleeger. "Principles of Survey Research Part 5: Populations and Samples". In: *ACM SIGSOFT Software Engineering Notes* 27.5 (Sept. 2002), p. 17. ISSN: 01635948. DOI: 10.1145/571681.571686. URL: http://portal.acm.org/citation.cfm?doid=571681.571686.

[19] Barbara Kitchenham and Shari Lawrence Pfleeger. "Principles of Survey Research Part 6: Data Analysis". In: *ACM SIGSOFT Software Engineering Notes* 28.2 (Mar. 2003), p. 24. ISSN: 01635948. DOI: 10.1145/638750.638758. URL: http://portal.acm.org/citation.cfm?doid=638750.638758.

[20] John Lewis. "Redefining Qualitative Methods: Believability in the Fifth Moment". In: *International Journal of Qualitative Methods* 8.2 (June 2009), pp. 1–14. ISSN: 1609-4069. DOI: 10.1177/160940690900800201. URL: http://journals.sagepub.com/doi/10.1177/160940690900800201.

[21] Lucy Ellen Lwakatare, Pasi Kuvaja, and Markku Oivo. "Dimensions of DevOps". In: *Lecture Notes in Business Information Processing*. Vol. 212. 2015,

pp. 212–217. ISBN: 9783319186115. DOI: 10.1007/978-3-319-18612-2_19. URL: http://link.springer.com/10.1007/978-3-319-18612-2_19.

[22] Joseph Maxwell. "Understanding and Validity in Qualitative Research". In: *Harvard Educational Review* 62.3 (Sept. 1992), pp. 279–301. ISSN: 0017-8055. DOI: 10.17763/haer.62.3.8323320856251826. arXiv: 0017-8055. URL: http://hepgjournals.org/doi/10.17763/haer.62.3.8323320856251826.

[23] Michael D Myers and Michael Newman. "The qualitative interview in IS research: Examining the craft". In: *Information and Organization* 17.1 (Jan. 2007), pp. 2–26. ISSN: 14717727. DOI: 10.1016/j.infoandorg.2006.11.001. URL: http://linkinghub.elsevier.com/retrieve/pii/S1471772706000352.

[24] Michael Quinn Patton. *Qualitative research & evaluation methods*. 3rd. London: Sage Publications, 2002, p. 598. ISBN: 0-7619-1971-6. URL: http://psycnet.apa.org/record/1990-97369-000.

[25] Shari Lawrence Pfleeger and Barbara a Kitchenham. "Principles of Survey Research Part 1: Turning Lemons into Lemonade". In: *ACM SIGSOFT Software Engineering Notes* 26.6 (Nov. 2001), p. 16. ISSN: 01635948. DOI: 10.1145/505532.505535. URL: https://dl.acm.org/citation.cfm?doid=505532.505535.

[26] Puppets labs & IT Revolution. *2015 State of DevOps Report*. 2015. URL: https://puppet.com/resources/whitepaper/2015-state-devops-report.

[27] QSR International. *NVivo*. 2018. URL: https://www.qsrinternational.com/nvivo/home (visited on 05/21/2018).

[28] Per Runeson et al. *Case Study Research in Software Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Inc., Mar. 2012, p. 237. ISBN: 9781118181034. DOI: 10.1002/9781118181034. URL: http://doi.wiley.com/10.1002/9781118181034.

[29] Forrest Shull, Janice Singer, and Dag I.K. Sjøberg. *Guide to Advanced Empirical Software Engineering*. Ed. by Forrest Shull, Janice Singer, and Dag I. K. Sjøberg. London: Springer London, 2008, pp. 1–388. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/978-1-84800-044-5.

[30] SurveyMonkey. *SurveyMonkey*. 2018. URL: https://www.surveymonkey.com/ (visited on 05/22/2018).

[31] Porcia Vaughn and Cherie Turner. "Decoding via Coding: Analyzing Qualitative Text Data Through Thematic Coding and Survey Methodologies". In: *Journal of Library Administration* 56.1 (Nov. 2015), pp. 41–51. ISSN: 15403564. DOI: 10.1080/01930826.2015.1105035. URL: http://www.tandfonline.com/doi/full/10.1080/01930826.2015.1105035.

[32] John Willis. *DevOps Culture (Part 1) - IT Revolution*. 2012. URL: http://itrevolution.com/devops-culture-part-1/ (visited on 03/07/2018).

[33] Robert K. Yin. *Case study research : design and methods*. 4th. Sage Publications, Inc, 2009, p. 219.

7

Appendix

A Interview Guide

Table 7.1 shows the interview guide that was used during the semi-structured interviews. Note that these questions were only the spring board for leading the interview. If other questions were considered relevant, they were brought up during the interview.

Table 7.1: Table showing the interview guide that was used during the semi-structured interviews.

#	**Questions**	**RQ**	**Rationale**
Background, context			
1	Can you introduce your work at <company >?		Context
1.1	Follow up on specific role and how long he/she has worked with DevOps related questions.		Context
1.2	Follow up on the kind of product and organization. - COTS / service etc. - Large / small		Context
2	How come you ended up where you are right now?	1	Context
Defining DevOps, and the Agent's DevOps related work			
3	How have you worked with DevOps at <company >?	2	Eliciting practices
4	What is your point of view in what it is to work with DevOps?	1, 2	Eliciting practices
4.1	What did you start implementing when you introduced DevOps in <company >?	2	Eliciting practices
4.2	What do you consider to be key components when implementing DevOps?	2, 3	Eliciting practices
3.1.1	Do you consider **x** to be part of DevOps?	2, 3	Eliciting practices
	List of suggestions of subject areas for **x**: - Culture: make Dev and Ops talking - Automation - Measuring, Monitoring, Sharing - Lean - (Micro)services		Helpful suggestions for question 3.1.1

#	**Questions**	**RQ**	**Rationale**
	Measuring DevOps practices		
5	What consequences have you experienced implementing DevOps?	4	Find out the effects (both the positive and negative)
	Ask about specific areas (if not asked about): - cultural (how are the people doing, how is it to work in the team, ...) - external (business goals, customer satisfaction) - internal (bugs, code quality, etc.) - process speed (LTD, MTTR, Deploy Frequency, ...)		Find out impact on different key areas
5.1	(If only the positive/negative ones are highlighted): What negative/positive consequences have you experienced?	4	Find out the non-mentioned effects
5.2	How did you find out that these consequences had taken place? (Tip: taken in impressions from the group, made measurements)	4, 5	Understand if it is subjective or objective
5.2.1	If measurements: How have you measured the changes?	5	Obtain metrics
5.2.1.1	Do you have any measurement data on these changes available?	5	Obtain measurements
5.2.2	If no measurements: What do you think could be measured?	5	Obtain metrics
5.3	Who makes sure you reach the desired result?		Validate knowledge of why metrics are useful
	Wrapping up		
6.1	If we want to evaluate if DevOps is worth implementing, what do you think is a good way of doing it?		Receive their learnings, and wrap up
6.2	What do you think is most controversial when it comes to convincing companies about implementing DevOps?		Receive their learnings, and wrap up

B Thematic Codes

Agile
Barriers
Business Value
Code Quality
DevOps Consequences
-> Negative DevOps Consequences
-> Positive DevOps Consequences
DevOps Definition
DevOps Impediments
DevOps Implementation
DevOps Practices
-> Automation
-> -> Automatic Deployment
-> -> Automatic Testing
-> -> Continuous Integration
-> -> Continuous Delivery
-> -> Continuous Deployment
-> -> Delivery Pipeline
-> Culture
-> -> Goal Sharing
-> -> Goal Sharing -> Definition of Done
-> -> Knowledge Sharing
-> -> -> Community of practice
-> -> Shared Responsibility
-> -> Task Sharing
-> -> -> Shared Roles
-> Fail fast
-> Feature Toggling
-> Feedback Loops
-> -> Hypothesis-driven Development
-> Infrastructure as Code
-> Leadership
-> -> Clear DevOps Vision
-> -> Support from Leadership
-> Monitoring
-> Services
-> Team Structure
-> -> Autonomous Teams
-> -> Cross-functional Teams
-> -> Pioneer DevOps Team
-> -> Supporting DevOps team
-> Test Driven Development
-> Testing

-> Value Stream
-> -> LEAN
-> -> -> Kanban
-> Versioning
-> Virtualization
DevOps Prerequisites
DevOps Topology
ERP Systems
Goals of Implementing DevOps
Measurement
Metric
Product Life Cycle
Quality Assurance
Recode everything
Security
What makes DevOps worth the effort

C Survey Questions

In this appendix, the questions from the survey along corresponding response options are stated in each section.

C.1 Question 1: In contexts where I have adopted DevOps, the following goals have been important

- Increase Throughput — Deliver faster, reduce time to market
- Increase Value of Delivery — Deliver the right thing
- Increase Product Quality
- Improve Work Environment — Employee satisfaction, Reduce deployment pain, Remove boring tasks, ...
- Increase Organizational Scalability
- Increase Product Scalability
- Reduce Truck Factor — Reduce dependency on certain people

For each of the practices, the respondent could choose one of the following alternatives:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree
- No opinion

C.2 Question 2: To reach my DevOps adoption goal(s), I have considered the following practices important

- Continuous Delivery — Working towards continually deliver deployment ready code. Working towards all code being releasable at any point.
- Continuous Deployment — Working towards continually deploy into production. Working towards all code being released into production immediately.
- Infrastructure as Code — Having the infrastructure declarative in source files, such that the setup of the infrastructure is repeatable.
- Automatic Testing — Working towards enabling automated testing of all code.
- Automatic Deployment — Automating the deployment of releasable code into the production environment.
- Continuous Integration — Continually integrate code branches into the master branch.
- Goal Sharing — Dev and Ops more explicitly share the same goal of meeting the need of the customers. This could be through a collaboratively defined

definition of done between Dev and Ops, a cross-functional team that together takes responsibility for Dev and Ops tasks and so on.
- Knowledge Sharing — Working towards sharing knowledge and learning from each other. This can be done by establishing communities of practice, by having Dev and Ops coach each other and so on.
- Shared Responsibility — Working towards Dev and Ops having shared responsibilities. This can look like Dev taking responsibility for what they deliver as it is deployed and running, for example when the software crashes Dev gets involved to solve and learn instead of brushing it of as Ops responsibility.
- Loosely-coupled Architecture — Organize the architecture to avoid monoliths, using for example a microservices design.
- Hypothesis-driven Development — Let the development be oriented on the experimentation of hypotheses. Set up a hypothesis, test it on all or some selected users and observe the results.
- Versioning of all Software Artifacts — Working towards version controlling everything: source code, configuration files, infrastructure, dependencies, etc.
- Fail-fast — Working towards reducing fear of making mistakes and instead embrace learning from mistakes. Deploy, and then fix the detected errors swiftly by deploying a new fix. This contrasts with the thinking where failures late in the cycle are expensive and must be avoided at all cost.
- Clear DevOps Vision — Leadership has established and clearly communicates the vision of going towards DevOps.
- DevOps adoption backed up by leadership — Leadership supports the process of adopting DevOps.
- User behavior Monitoring — Work towards monitoring how the users are using the system to provide feedback to Dev.
- System Monitoring — Monitor how the system is doing (performance, downtime, ...). This can be done on different levels (infrastructure, features, ...).
- Cross-Functional Teams — Working towards having all the knowledge needed for both Dev and Ops tasks in the same team. The team takes responsibility from requirements, to deployment to maintenance and support. Sometimes this is further broadened to include BAs, QA, etc.
- Broaden the Developer Role — Working towards Dev embracing tasks that traditionally belongs to other roles. Areas normally include making Dev more responsible for testing, deployment and support.
- Pioneer DevOps Team — Start adopting DevOps by creating a "shining example" team that other teams can resemble.

For each of the practices, the respondent could choose one of the following alternatives:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

- No opinion

C.3 Question 3: I have experienced the following effects from adopting DevOps

- Increase Throughput — Deliver faster, reduce time to market
- Increase Value of Delivery — Deliver the right thing
- Increase Product Quality
- Improve Work Environment — Employee satisfaction, Reduce deployment pain, Remove boring tasks, ...
- Increase Organizational Scalability
- Increase Product Scalability
- Reduce Truck Factor — Reduce dependency on certain people

For each of the practices, the respondent could choose one of the following alternatives:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree
- I don't know