DevOps in Finland - Study of Practitioners' Perception

University of Oulu
Faculty of Information Technology and Electrical Engineering / Information Processing Science
Master's Thesis
Tuomas Paulin
November 28, 2018

Abstract

DevOps is currently one of the latest software development practices. Lately it has gained the interest of people in academia and practice. DevOps extends Agile practices to software operations and aims to make software development process faster, more reliable and increase collaboration. Currently there are multiple studies which aim to define DevOps but only a few which try to understand and evaluate how DevOps is utilized and understood in practice and at large.

The aim of this study is to investigate DevOps adoption, practices and tool usage by software professionals in Finland. In addition, the study investigates perceived benefits and challenges of DevOps adoption.

A survey with an online questionnaire was selected as the method for gathering data from software practitioners in Finland. Previous literature focusing on DevOps was used to establish an understanding of DevOps and to create meaningful questions for the survey. A link to online survey questionnaire was then distributed using Slack, LinkedIn and mailing lists during Spring 2018 to Finnish practitioners. Multiple channels were selected to collect sufficient responses for analysis. A total of 81 respondents answered to the questionnaire and were from different backgrounds with respect to organization size, role and team size.

Most of the participants had already adopted DevOps and clear understanding of the concept was considered the most important factor in DevOps implementation. Automation was both an important meaning of concept and also most agreed practice. Faster release cycle time and system quality were the most agreed benefits and lack of common understanding for DevOps was considered the most challenging. A multitude of different tools are used in organizations. The most popular in their own categories were Jenkins(CI), Kibana(Monitoring), Amazon AWS(Cloud) and Ansible(Config/Provisioning).

Automation was considered important aspect of the DevOps concept and also in practice. Further research and qualitative data is required to find out the actual reasons behind these results. The questionnaire instrument can be reused on different target groups. Qualitative questions should be asked on organization level to find out the reasons behind different implementations of DevOps.

Keywords
Software development methods, DevOps, Survey, Tools

Foreword

This thesis was started in Autumn 2017 after my bachelor's thesis was completed. I was encouraged by my previous supervisor, Raija Halonen, to contact Lucy Ellen Lwakatare, Ph.D. based on my interests in DevOps and modern software development methods. The process started very quickly after that. After a couple of meetings with Professor Pasi Kuvaja and PostDoc Lucy Ellen Lwakatare, and I started to plan and conduct the study with their guidance.

I want to thank my family, friends and my supervisors for helping me get through this process. Special thanks to Satu Inkinen, Ph.D. for providing comments on the raw version.

In Brno 28.11.2018

Tuomas Paulin

Abbreviations

DevOps - Development and Operations

CI - Continuous integration

CD - Continuous Deployment

VSTS - Visual Studio Team Services

TFS - Team Foundation Services

IaaS - Infrastructure as a Service

PaaS - Platform as a Service

AWS - Amazon Web Services

SD - Strongly Disagree

D - Disagree

N - Neutral

A - Agree

SA - Strongly Agree

Contents

1 Introduction

Software development has come a long way in the last four decades. Different languages, processes and methods have been used throughout the years. DevOps is among the popular modern software development approaches. DevOps as a term is a combination of software development(Dev) and operations(Ops). The term was coined in 2008 (Debois, 2008) and has been studied since by multiple people in academia and practice. DevOps uses the principles of Agile software development and extends the focus of collaboration and practices to operations (Gupta, Kapur, & Kumar, 2017).

Several studies have attempted to define what DevOps means, resulting in multiple different definitions of the concept (Bass, Weber, & Zhu, 2015; de França, Jeronimo, & Travassos, 2016; Smeds, Nybom, & Porres, 2015; Stahl, Martensson, & Bosch, 2017; Dyck, Penners, & Lichter, 2015; Jabbari, bin Ali, Petersen, & Tanveer, 2016; Gupta et al., 2017). Other scholars have suggested that adopting DevOps requires different activities for companies. These activities include changes to organizational structures (Shahin, Babar, Zahedi, & Zhu, 2017; Nybom, Smeds, & Porres, 2016), system architecture using microservices (Balalaie, Heydarnoori, & Jamshidi, 2016) and utilizing a completely automated deployment pipeline (Bass et al., 2015). Some studies have highlighted benefits and challenges of DevOps and its adoption. Benefits include faster development of software (Riungu-Kalliosaari, Mäkinen, Lwakatare, Tiihonen, & Männistö, 2016; de França et al., 2016), better quality (Riungu-Kalliosaari et al., 2016; de França et al., 2016), enhanced collaboration and communication (Riungu-Kalliosaari et al., 2016; de França et al., 2016; Nybom et al., 2016) and challenges like insufficient communication (Riungu-Kalliosaari et al., 2016), cultural problems (Riungu-Kalliosaari et al., 2016; de França et al., 2016; Nybom et al., 2016), management problems (Jones, Noppen, & Lettice, 2016), aligning of existing processes (de França et al., 2016) and hardware dependency (Lwakatare et al., 2016).

Motivation for this study is to find out the use of DevOps in Finnish software industry and see if research and practice are currently having similar thoughts (or well aligned) with regards to DevOps. Combining benefits, challenges, tools and practices should give insights into how DevOps is currently perceived and where the potential problems lie. At the time of writing, there were very few exploratory survey studies done in this subject. Specifically, in Finland, multiple case studies have been conducted showing increasing adoption and interest on DevOps. However it is not clear how the different claims in literature about DevOps are perceived at large by different professionals in ICT.

The aim of this study is to collect relevant existing research in DevOps area, investigate practitioners' perception of DevOps adoption, usage and tools at large using survey research approach and to contribute to current DevOps research using a survey research method. The main research question of this thesis is: **What is the state of DevOps in Finland?** The following research questions were created to answer the main question.

- **RQ1**. What is the perception of DevOps adoption and use by Finnish ICT professionals?
- **RQ2**. What engineering practices and tools are applied in DevOps adoption among Finnish ICT professionals?

- **RQ3**. What are the perceived benefits and challenges of DevOps adoption by Finnish ICT professionals?

A literature review on concept and usage of DevOps is conducted to gather material for the survey, which should answer to second and third research question. The current state of DevOps in Finland is analysed using data from a self-administered questionnaire. The survey was selected as the main data gathering method and the questions are based on existing literature found in the literature review and by using one non-academic questionnaire as a starting point for questions (Brown, Forsgren, Humble, Kersten, & Kim, 2016). The focus group of this survey and questionnaire was software industry personnel working in companies in Finland. The results from questionnaire provide explanations to the research questions provided earlier.

The contributions of this paper are as follows. First, the empirical data collected sheds light on the practicioners' point of view on concept of DevOps, its benefits and challenges, in addition with tools, already discussed in literature. Second, the questionnaire created is a good starting point for other studies to start investigating the adoption of DevOps in other target groups.

This thesis contains six main chapters. In the next chapter, the results of the literature review are presented and summarized. Explanation of the survey method, design, implementation and analysis are presented in the research methods in chapter 3. The results from the self-administered questionnaire are presented in the results chapter 4 with tables and figures split into categories used in the questionnaire. In discussion chapter 5 the results and their implications are discussed and limitations of this study are described. Finally, conclusion in chapter 6 includes final words on the implications followed by suggestions for further research.

2 Background and related studies

This chapter presents different aspects of DevOps including definition, concept, practices, benefits and challenges as they are presented in the literature. Multiple proposed definitions of DevOps are presented in the first sub-chapter. Second sub-chapter focuses on adoptions and implementations of DevOps. Finally, third sub-chapter includes benefits and challenges found in the literature.

2.1 DevOps

DevOps as a term has existed for almost 10 years. First introduction was in 2008 by Patrick Debois (2008) in his presentation about agile infrastructure and methods in organisations. There are multiple different aspects to DevOps. DevOps can be seen as a set of practices (Bass et al., 2015), a mindset that aims to increase collaboration (Rajkumar, Pole, Adige, & Mahanta, 2016) and it can be divided into different elements, such as collaboration, automation, measurement and monitoring (Lwakatare, Kuvaja, & Oivo, 2015).

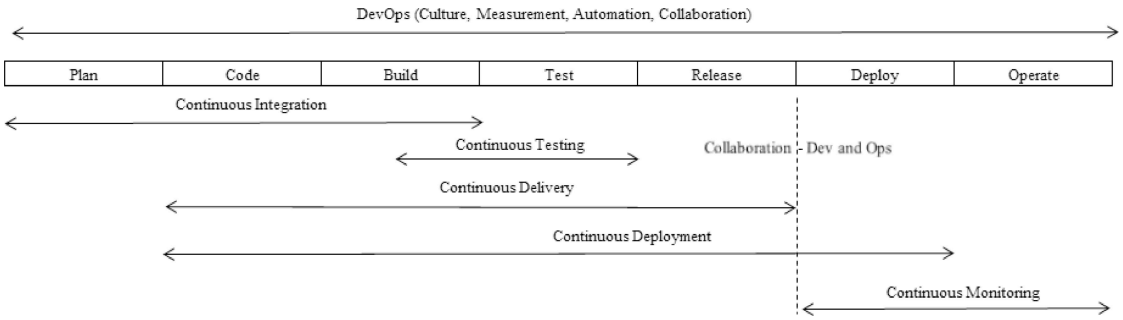
A multivocal literature review was conducted by de França et al. (2016) where academic literature, industrial reports, tool vendors' websites and blog posts were used to grasp a thorough insight on DevOps. The authors wanted to find out the meaning behind DevOps, issues motivating the adoption of DevOps, main characteristics and benefits and challenges of DevOps. de França et al. (2016) discarded DevOps as a method because of the lack of systematic approach to introduce or perform DevOps in organizations. Smeds et al. (2015) argue that DevOps is not a revolution but adopting it might require disruptive change. According to them DevOps contains cultural aspects that are critical but they are not defining aspects by themselves and definition of DevOps should include engineering practices co-existing with cultural factors. While clarifying the differences in definitions of DevOps and Release Engineering, Dyck et al. (2015) present an approved common ground with previous literature confirming that communication and collaboration are the backbone of DevOps and both development and maintenance of systems were seen as equally important.

Ståhl, Mårtensson and Bosch (2017) concluded that DevOps is a concept consisting of multiple parts similar to definition of Agile. DevOps is a combination of values, principles, methods, practices and tools. Values and principles are hard to pin down and do not help in defining the actual behaviour. Ståhl et al.(2016) argue that community should focus more on methods, practices and tools because they are easier to measure and more concrete than values.

Gupta et al., (2016) studied DevOps implementation using statistical analysis methods and proposed a framework to assess 18 different attributes to measure DevOps implementation. They also compare DevOps phases to the continuous "everything" methods that are required for realizing DevOps. Different phases of development included in the continuous methods are presented in Figure 1.

Systematic reviews of DevOps have been conducted by several scholars with the aim of improving the understanding of the concept. The practices of DevOps and differences

Figure 1: Continuous methods presented by (Gupta et al., 2017), licensed under CC BY-NC-ND 4.0



and similarities to other methods were also studied. Table 1 summarizes the multiple definitions of DevOps proposed in the literature.

As discussed in multiple articles, definition of DevOps seems to be a moving target and defining it concretely shares similar problems as defining Agile (Abrahamsson, Salo, Ronkainen, & Warsta, 2017; Dingsøyr, Nerur, Balijepally, & Moe, 2012). Elements of these definitions are used in survey creation and are tested in software industry to see how respondents view or feel about different aspects of DevOps.

Table 1: Definitions of DevOps by multiple authors

Authors	Definition
(Bass et al., 2015)	"DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality."
(de França et al., 2016)	"DevOps is a neologism representing a **movement** of ICT professionals addressing a **different attitude** regarding software delivery through the **collaboration** between software systems development and operation functions, based on a set of **principles** and **practices**, such as **culture**, **automation**, **measurement** and **sharing**."
(Smeds et al., 2015)	"DevOps is a set of **engineering process capabilities** supported by certain **cultural** and **technological** enablers."
(Stahl et al., 2017)	DevOps is a combination of values, principles, methods, practices and tools. Focus is on practices and tools (No strict definition).
(Dyck et al., 2015)	"DevOps is an organizational approach that stresses **empathy** and cross-functional **collaboration** within and between teams – especially development and IT operations – in software development organizations, in order to operate resilient systems and **accelerate delivery** of changes."
(Jabbari et al., 2016)	"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."
(Gupta et al., 2017)	"DevOps is a set of principles that advocates strong integration and information sharing between the development and operation teams. The approach heavily relies upon maximum automation across the entire life cycle of software development, deployment and operations. The key aim of DevOps is to maximize the **predictability**, **efficiency**, **quality** and **productivity** of the development and operations. The guiding principles behind DevOps are **culture**, **measurement**, **collaboration** and **automation**."

2.2 Adoption and implementation of DevOps

Global surveys on the State of DevOps have been conducted annually by Puppet since 2013 (Brown et al., 2016). These surveys provide useful information and a global overview of the state-of-the-practice due to the large number of respondents, but have questionable scientific rigour [1]. From scientific venues, two surveys studied DevOps directly (Shahin, Babar, et al., 2017; Shahin, Zahedi, Babar, & Zhu, 2017). One study included interviews and a quantitative survey which had respondents across the globe (Cito, Leitner, Fritz, & Gall, 2015). In contrast to this study, these surveys have a global perspective instead of a narrow focus on Finland.

Most empirical studies of DevOps adoption and implementation use different research approaches but majority use case study approach (Nybom et al., 2016; Smeds et al., 2015; Jones et al., 2016; Lwakatare et al., 2016). From the empirical studies insights about DevOps usage across different organisation types can be gained, such as insights into practices (Shahin, Babar, et al., 2017) and challenges (Smeds et al., 2015; Jones et al., 2016). The insights are used to create questions during the survey design process.

Controlled software development in an organization usually requires certain rules and sets of tools to be in place. Evaluation of such software development can be achieved by using a framework called BAPO (Business, Architecture, Process, Organization). These aspects have their own levels but additionally also flow naturally starting from business and ending to the organization. Business is the ability to manage the costs and profits of development, architecture considers the common parts of software (product families) and the reusability of components, process relates to the process, roles, work products and responsibilities in development and organisation is the ability to deal with complex relationships and responsibilities. Changes in any of these aspects will induce changes in other aspects and focusing in one of the dimensions can result in downgrades in others if they are not considered. (Van Der Linden, Bosch, Kamsties, Känsälä, & Obbink, 2004).

Bucena and Kirikova (2017) propose a similiar model as BAPO for DevOps, which gives different stages of implementation consisting of levels 1-5 where 1 indicates mostly manual work and 5 indicates automated processes in software development. The considered areas of enterprise are technology, process, people and culture. Organisation can select what levels are feasible for them and then aim to achieve and measure them accordingly. Approach includes measuring current software maturity level, selecting desired DevOps level and detecting current impediments that need to be tackled. Afterwards required practices and tools need to be listed and the first improvement planned. The DevOps adoption includes choosing the object, identifying metrics to measure and starting the actual adoption phase during which the results are collected and shared accordingly. After this is done, next phase is to either select another improvement, start the whole process from beginning, or to stop the adoption process. Table 2 contains three examples from the technology area of the proposed model (Bucena & Kirikova, 2017).

The framework was tested in a medium-sized organization which had a small IT department with silos (operations and three separate development teams). The company had already implemented DevOps but there were problems with the practices not being implemented completely. DevOps practices and knowledge were lacking and there

[1] Puppet is a company that develops Puppet Enterprise, a commercially supported version of the Puppet open-source configuration management tool. Puppet is often considered a DevOps tool.

Table 2: An example of technological aspects of DevOps maturity model proposed by (Bucena & Kirikova, 2017)

Initial level (1)	Repeatable level (2)	Defined level (3)	Managed level (4)	Optimized level (5)
Environments are provisioned manually	All environment configurations are externalized and versioned	Virtualization used if applicable	All environments managed effectively	Environment provisioning fully automated
Manual tests or minimal automation	Functional test automation	Triggered automated tests	Smoked tests and dashboard shared with Op.t.	Chaos Monkey
Data migration unversioned and performed manually	Changes to DB done with automated scripts versioned with application	DB changes performed automatically as part of deployment process	DB upgrades and rollbacks tested with every deployment	Feedback from DB performance after each release

was high level of waste in build automation. Management got the information they needed after the framework was applied and the experiment was concluded as a success. (Bucena & Kirikova, 2017)

Adopting DevOps in an organisation requires certain aspects to be integrated. The core aspects are capabilities and cultural and technological enablers. These capabilities, such as continuous development, and enablers, such as shared goals and build automation together make the DevOps transformation complete. Capabilities can be seen as the main aspects of DevOps, and they can be achieved by relying on the enablers to work them out in the organization. (Smeds et al., 2015)

The adoption requires a certain mix of software developers and operations personnel to actually achieve full automation of the infrastructure. Nybom, Smeds and Porres (2016) studied how mixing responsibilities affects the culture, tooling and way of working. For the company culture, DevOps creates friction around the new tasks required from both parties, but at the same time it is an eye-opening experience for software developers to see the operations side. Positive aspects for the culture are shared responsibilities and improved collaboration. Tools used is also one aspect. The quality of the tools requires awareness as there are new deployment risks and also tooling obstacles in the way of full automation. Impacts on ways of working are new responsibilities, new administration rights for development to make it easier for developers to modify the environment without operations personnel, and also common ways of working were seen as important with mixed responsibilities. Potential risks and concerns with the mixed responsibilities included unpleasant tasks (people like to do what they usually do) and creating problematic solutions for the tasks that were normally operational. Configuration errors were also considered problematic since same configurations were altered by both software developers and operations personnel. Different persons reacted differently on the new tasks and some hated it. (Nybom et al., 2016)

A smaller organization in UK adopted DevOps, because their old software system was considered a legacy system and the new system was to be developed using new methodologies to ensure that it wouldn't be monolithic, poorly documented and difficult to maintain. Adoption of the DevOps was conducted by a new software development manager who created a new team to create a new system to replace the old system and also wanted the development and operations teams to start working in close collaboration. The old system was still in production while the new system was designed so the time of software developers and operations was an issue during the adoption. Software development manager had to undertake this role, and at the same time champion the DevOps approach for both development and operations employees, and later an adjustment was suggested that software development manager would manage both teams as a whole. (Jones et al., 2016)

University of Messina was a leading party in a DevOps-centric project that produced a Cloud testbed for a customer. They wanted to utilize the DevOps paradigm: promote communication, collaboration and integration between software developers and operations. For the solution, they wanted their infrastructure to be flexible and robust and created using open source tools. Secure connections, possibility to scale up and intelligent integration methods were also included as features. The final project was a very customized development platform that required the usage of DevOps due to its configuration and requirements. (Bruneo et al., 2015)

Another DevOps transformation happened in Australia when Wotif group, who run an e-commerce platform for travel, wanted to overhaul their software release process to get the software release time from weeks to hours. There were many problems in relation to the software process. The development teams struggled with delays and problems with releases and operations team was overloaded and constrained. The release process was bureaucratic and the releases needed to be booked in advance and were often overruled by final second priority changes. In addition, the releases needed to go through 2 days of performance for every change which resulted in more changes in one release which resulted in more unstable releases. The solution for these problems was achieved using DevOps methodologies and tools with continuous delivery. They created a pipeline that had rules and encouraged new principles. The rules were: keep changes independent, no manual testing, no release slots reservation, applications must comply with the standards and operations staff can roll back releases after hours. These simple rules with the new tools and principles fixed most of the problems and drove the release cycle time to about one day per release. (Callanan & Spillane, 2016)

Empirical studies in Finland also show the adoption of DevOps across organisations. In one study they highlight Continuous Deployment and Delivery (CD+CD) practices, different development processes and specific toolchain related to it in their study. Case company deals with Kanban and rapid – post-agile – deployment process for their customer, who use the CD pipeline to create proof-of-concept (POC) solutions to see if they are profitable and then decide if the project should be developed into a product or eliminated. (Leppänen, Kilamo, & Mikkonen, 2015)

One study in Finland focused in tools and toolchains: what toolchains are used, what are the reasons for not using certain tools and how do they affect the speed of delivery. Interviews were conducted to 18 case organizations and they were further grouped based on themes (thematic analysis). Five phases of development were used: requirements, development, operations, testing and quality. Over 100 tools were identified in total and

all case organizations didn't have tools for every phase. The differences in products also meant that some categories were irrelevant for some of the companies. Faster deployments were enabled with complete toolchain but still a good toolchain wasn't the only reason for fast deployments. (Mäkinen et al., 2016)

In another set of interviews 15 companies were contacted to find out what makes CD beneficial and what are the challenges currently. Benefits found were faster feedback, more frequent releases, improved quality and productivity, improved customer satisfaction, effort savings and closer connection between development and operations. The challenges indicated that full automation was not yet achieved for the whole development chain. Deployment capability was an obstacle for on-demand deployments and continuous deployments. The company/product type played a part in this as well since not all companies strived for full continuous deployment due to domain problems, regulations and customer preferences – Some customers preferred slower releases. (Leppänen et al., 2015)

Adopting and implementing DevOps in a company requires that there are requirements for faster implementations and shorter (and more efficient) projects (Soni, 2015). The current situation needs to be analysed thoroughly to plan the required changes to improve either piece by piece (Bucena & Kirikova, 2017) or actually enable certain aspects to work (Smeds et al., 2015). The actual implementation process depends on the needs, problems and requirements generated by the customers (Shahin, Babar, et al., 2017), employees (Jones et al., 2016) and even the project or product (Bruneo et al., 2015) of the company. The tools play a huge role when the higher automation levels, monitoring and communication are to be achieved (Stahl et al., 2017; Shahin, Babar, et al., 2017). The cultural and technical issues before and during the adoption need to be dealt with to ensure smoother transition to DevOps culture, practices and ways of working (Smeds et al., 2015). Measurement related to the adoption should be defined to find out if the changes resolve the current issues (Di Nitto, Jamshidi, Guerriero, Spais, & Tamburri, 2016) and measurement is also an integral element in DevOps (Lwakatare et al., 2015; Gupta et al., 2017; Jabbari et al., 2016). The practices of DevOps are summarised in Table 3.

2.3 Benefits and challenges of DevOps

DevOps has recently been implemented in projects that are struggling with issues in traditional waterfall type projects, where the deployment happens only when the product is ready to launch. While definition of DevOps is quite well-defined, different companies take different aspects of it to their own toolkits. Agile practices and DevOps culture help the organization to achieve faster time to market and efficient outcomes. (Soni, 2015)

Geographical distribution, buzzword tiredness and more work and knowledge are problematic in the cultural aspect. It is difficult to work in distributed teams and share the same values over multiple time zones and cultures. Some personnel interviewed expressed distrust towards DevOps being a buzzword which brings negativity to the concept. Acquiring knowledge and working in the "other side" of Dev/Ops is seen as more work for the people that are not familiar with the other side and some people aren't interested in the other side at all. Belonging in either group was seen as more beneficial than belonging in neither group. In technologies problems can occur with software architecture, mismatch of environments (development vs production) and multiple production

environments. Monolithic architecture is a problem for fast releases and restructuring and rewriting is needed to get over it. Environment mismatch is a problem for releases and can be averted using technological enablers. (Smeds et al., 2015)

Soni (2015) studied the current challenges of insurance industry and created a proof-of-concept solution that would solve their current challenges in software development processes. They were struggling with customer expectations, manual build processes, lack of verification and validation of builds and slow application release process. The resulting stack of tools and techniques was tailored for the requirements. (Soni, 2015)

One of the reasons for using DevOps are problems in software deployment chain. Lwakatare et al. (2015) present in their systematic literature review four problems that DevOps solves: poor communication, manual operations processes, performance of development and quality assurance (QA) and monitoring data is segregated. Outcomes of DevOps include: shared responsibilities, continuous deployment of functionality, measurement of performance using operational data and feedback as consolidated view of operational data. (Lwakatare et al., 2015)

The problems of DevOps implementations were studied by Kamuto (2017) in South African context. Interviews were conducted to find out the factors that were hindering the adoption of DevOps in relation to existing studies on DevOps implementations. Five (5) inhibitors were introduced and possible interventions were planned to counter the harmful effects. Cultural change required was also discussed, but it is not the focus of this study. (Kamuto & Langerman, 2017)

Jones et al. (2016) identified management and team resistance as a challenge in the study conducted in small organization in UK. This was identified by the lack of business analysis in adoption of DevOps and the reluctance of management to allocate resources for the cause. Another problem indicated was the gap between current development and operations teams caused by different roles and management structure. (Jones et al., 2016)

DevOps practices were discussed earlier and automation in software deployment, and infrastructure management was highlighted by most scholars. Benefits like faster delivery and automated deployments were found in multiple studies. Most popular challenges identified were lack of knowledge and education, in addition to organizational structure and resistance to change. (Table 3)

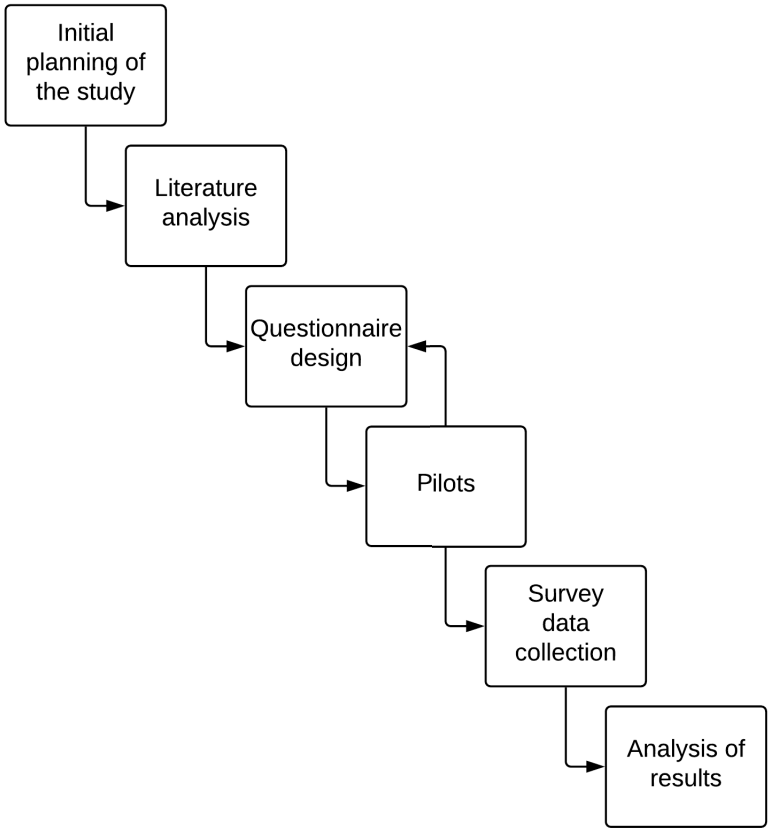
Table 3: Practices, benefits and challenges of DevOps identified by authors

Devops practice / benefit / challenge	**Authors**
Practices	
Automation of software deployment mechanisms	(Callanan & Spillane, 2016; Cito et al., 2015; Leppänen et al., 2015)
Automation in infrastructure management	(Callanan & Spillane, 2016; Leppänen et al., 2015)
Continuous monitoring by development team	(Cito et al., 2015)
Self-infrastructure provisioning by development team	(Leppänen et al., 2015; Nybom et al., 2016)
Reorientation of development and operations team structures and responsibilities	(Nybom et al., 2016; Balalaie et al., 2016)
Benefits	
Rapid or faster delivery	(Balalaie et al., 2016; Soni, 2015; Rajkumar et al., 2016; Ebert, Gallardo, Hernantes, & Serrano, 2016)
Automated deployments	(Soni, 2015; Jabbari et al., 2016)
Improved software quality	(Callanan & Spillane, 2016)
Challenges	
Lack of education or knowledge of DevOps	(Kamuto & Langerman, 2017; Balalaie et al., 2016; Smeds et al., 2015; Bucena & Kirikova, 2017; Riungu-Kalliosaari et al., 2016)
Resistance to change in management, team or organization level	(Kamuto & Langerman, 2017; Jones et al., 2016; Shahin, Babar, et al., 2017; Rajkumar et al., 2016)
Organizational structure, roles or culture	(Kamuto & Langerman, 2017; Smeds et al., 2015; Riungu-Kalliosaari et al., 2016; Lindgren & Münch, 2016)

3 Research methodology

This chapter explains the theory behind the selected research method and its actual implementation in this thesis. Survey research method with a questionnaire is introduced including its design and evaluation and presented in the first two sub-chapters. Implementation and analysis of responses are presented in subsequent two sub-chapters. The whole process is visible in Figure 2.

Figure 2: Research process



3.1 Survey and literature analysis

Survey was selected as the main research method for this thesis to investigate the overall state of DevOps from multiple ICT professionals in Finland. A literature review was conducted in order to create a relevant survey that uses right measures. The latter was in addition to identifying previous questions that have been inquired by previous survey studies related to DevOps and other types of studies related to DevOps adoption, implementation and benefits and challenges.

Survey is a method for gathering data that is to be analysed from a source. A good survey has to reach a large number of respondents, represent a large population and create standardized, quantifiable, empirical data. Creating a perfect survey is a difficult task which requires planning, testing and verifying the questions. Basic survey types include descriptive, explanatory, census, cross-sectional, trend, panel study surveys. Surveys can be conducted face-to-face, via telephone and be self-administered. The selection is based

on what is considered appropriate for the study and how the target group is considered to be contacted. The survey process has many steps including planning the approach, constructing the questionnaire and cover letter, conducting the pilot, redeveloping, executing the actual survey after which the analysis of data begins. (O'leary, 2004)

3.2 Survey design and evaluation

The process started by looking at existing surveys with self-administered questionnaires which were found in the literature review and one previously selected non-academic questionnaire: State of DevOps 2016 (Puppet, 2016). Relevant case studies of DevOps implementation and literature reviews were also inspected for possible new questions. The sources collectively served as inputs to the list of questions. Google Sheets was used to initially populate and manage the list of questions that were structured and described with respect to question type, options and rationale based on the prior literature. For questions measuring opinion, Likert scale from 1–5 (strongly disagree (1), disagree(2), neutral(3), agree(4), strongly agree(5) was used in the questionnaire. The questionnaire consisted of both closed and open-ended questions developed iteratively and evaluated twice through pilot tests administered on Google forms.

The first pilot questionnaire was submitted to a pre-selected pilot group from the industry and two students were approached at the University of Oulu for testing. Five professionals from industry that participated in the first pilot test were from DevOps Oulu workshop. The DevOps Oulu workshop was initiated and attended by several practitioners from two companies developing embedded systems. At the end of November 2017, I participated in the workshop, presented some information about survey study and shared a questionnaire link. Each of the questions had their own comment field to enable self-administered feedback for the industry group and the students were monitored while answering and asked to think aloud when filling the questionnaire. Multiple adjustments were done after the first pilot to make the questionnaire more lenient to respondents.

Second pilot questionnaire was created using the results from the first one and tested by two students working in the industry. Their feedback was used to improve the wording of the questions and resulted in the removal of two problematic questions.

The most important improvements to the pilot questionnaire were to reduce the answering time and to create paths for different roles of respondents. Questions were cut from 60 to 30 to resolve the issue with time and a couple of branching questions were added to make the questionnaire as a whole easier to answer with people from different backgrounds.

The final questionnaire was moved from Google Forms to Webropol survey platform and restructured according to the feedback received from the pilot tests. The final questionnaire had a total of 30 questions divided into four parts: background, DevOps, development and tooling. Questions in the background category captured the size of the organisation and role of the respondents. The DevOps category explored the adoption of DevOps, including respondent perceptions about the definition of DevOps, initiatives considered during the adoption, benefits and challenges. Detailed respondent experiences with DevOps were captured in the development category, which also allowed

respondents to proceed or alternatively submit their response. Tooling was the last category in the survey and allowed respondents to select answers about the tools used to support the DevOps approach. At the end of the survey, respondents were given an option to provide comments, feedback or contact details to send survey results. The final list of questions from the questionnaire can be found in the end of this study (Appendix A).

3.3 Implementation

The questionnaire was distributed as a simple link using different cover letters to multiple software development professionals in Finland using email, Slack and LinkedIn as the sharing channels. It was advertised in two Slack groups: Koodiklinikka[2] and DevOps Finland[3]. Additionally the questionnaire was advertised using a public status update that was shared by multiple people and private messages to some of the people who were in DevOps group in Linkedin. No identifying information was collected during the survey but the respondent was allowed to leave their e-mail in case they wanted to receive the results submitted to them personally. Different sharing methods are displayed in the table below. (Table 4)

Table 4: Sharing of survey link

Type	**A**. Description (Number of shares) **B**. Description (Views / channel size)
A. Sharing via E-mail	
Email	Emails sent to contacts (18 addresses)
Mailing List	FisMA (150), DevOps Finland Meetup (1706), DevOps Oulu workshops (31)
B. Sharing via social media	
Slack groups	DevOps Finland Meetup (160 people), Koodiklinikka (1168 people)
LinkedIn	Private message (10), public post (269 views)

The online questionnaire was made publicly accessible from January to April in 2018 and the respondents were connected in waves using the methods specified above. Most of the responses were collected in April 2018.

3.4 Analysis of responses

The results of the questionnaire were analyzed by using quantitative research methods. Quantitative research is used in analysing data collected from a designed instrument(s). The approach requires numerical data from the selected empirical research

[2]Koodiklinikka is a Finnish support group for programming (*Koodiklinikka*, n.d.)

[3]DevOps Finland is an open group of people interested in DevOps and everything around it.(*DevOps Finland*, n.d.)

method. Quantitative methods are used in research to establish general laws or principles (Burns, 2000). For an excellent survey, good coverage of questions, a solid sample of people, a lot of responses and valid measurement methods need to come together to form a great specification (Leeuw, Hox & Dillman, 2007). Quantitative research is a process which includes all of the stages of academic research process, including literature review, theory development and design and execution stages. The motivation of researcher using quantitative research lies in the numerical outputs and how to acquire meaning from it using statistical methods. (Straub, Gefen, & Boudreau, 2005).

The basic approach of quantitative research includes selecting the type of research, general research approach, data collection technique and data analysis technique. Type of research is either confirmatory or exploratory research. Confirmatory research aims to test existing relationships and exploratory tries to define possible relationships. Common general research approaches include case studies, field studies etc. One or more data collection techniques can be used. These include, for example, surveys, observations and secondary data sources. Data analysis techniques include simple and complex statistics methods that aim to uncover different aspects of the collected data. (Straub et al., 2005).

SPSS was the primary tool in analysis of the results. Analysis technique was selected accordingly depending on the question. Frequencies were calculated for every question. For the Likert-scale question, central tendency, (median–Mdn) and dispersion (interquartile range–IQR) measures were determined to assert common opinions among respondents and the spread of other opinions. In addition, agree-strongly agree and disagree-strongly disagree options were combined together to compare rates of agreement and disagreement in textual format in some cases. Two independent background variables, company size and role, were used to get more insight on different variables are affected by them.

A couple of respondents did not answer all questions – despite of making the questions required to be answered – and those results are marked with different totals(N) than the others in the next chapter. Questions 1, 2 and 26 – 30 were recoded in order to combine results from multiple questions. In the first question, the ranges were reduced from five to three (small, medium, large) and in the second question most prevalent “other” roles were added as their own categories. In the final four questions on tools, most common “other” tools from each tool category were also included to the results.

4 Results

The results from the questionnaire are presented in this chapter. The sub-chapters are divided by the themes of the questions: background, DevOps opinions, technical details and tooling. The questionnaire had 81 respondents in total and 64 of them answered in technical questions.

4.1 Background of respondents

Two background questions were used as the independent variables which could be used in analysis of other questions. Size and working role were selected because they were perceived as easy to answer and would provide much needed data while at the same time keeping the questionnaire anonymous. In the technical section of the survey two additional questions were asked to find out the team sizes and application types.

First question of the background questions focused on size of the organization in which respondent was working currently. Most of the respondents were in small (1-100 employees, 41%), followed by medium (101-1000 employees, 32%) and large(1000+ employees, 27,2%) organizations. (Table 5)

Table 5: Organization size

Company size	**N**	**%**
Small (1−100 employees)	33	41%
Medium (101−1000 employees)	26	32%
Large (>1000 employees)	22	27%
Total	81	100%

This question was featured in other survey studies, either using different grouping (1-5, 6-10, 11-50, 51-150, 151-500 and 500+) (Garousi, Coşkunçay, Betin-Can, & Demirörs, 2015) or as a numeric question (Stankovic, Nikolic, Djordjevic, & Cao, 2013).

There were multiple different roles featured (Table 6). Most common from provided options was Software developer / engineer (33.3%), followed by software architect (17.3%) and third was operations / system administrator (14.8 %). 29,6% of respondents marked that they were other than the five options provided for this question. The biggest roles found in the ”Other” category were DevOps Architect, CTO (3), Agile Coach (3) and DevOps Specialist(2).

The roles were a popular variable in other questionnaires and interviews. Roles like manager, engineer, developer and other (specify) were used in each of the related studies (Rodríguez, Markkula, Oivo, & Turula, 2012; Shahin, Babar, et al., 2017; Garousi et al., 2015). Architect was used in three studies (Rodríguez et al., 2012; Shahin, Babar, et al., 2017; Garousi et al., 2015) and operations staff was used in two (Shahin, Babar, et al., 2017; Rodríguez et al., 2012). Quality assurance and testing was used in two studies as well (Rodríguez et al., 2012; Garousi et al., 2015).

Two additional background questions were featured in the technical section. Development team sizes were small. Two first categories of team sizes (1−5, 6−10) together had

Table 6: Roles of respondents

Role	N	%
Software Developer / Software Engineer	27	33.3%
Software Architect	14	17.3%
Operations staff / System administrator	12	14.8%
Quality Assurance / Test Engineer	2	2.5%
Project manager	2	2.5%
Other	24	29.6%
Total	81	100.0%

82.8% of responses while two largest team sizes had only single respondents featured. (Table 7)

Table 7: Development team sizes (N = 64)

Team size	N	%
1–5 persons	24	37.5%
6–10 persons	29	45.3%
11–20 persons	5	7.8%
21–50 persons	4	6.2%
51–100 persons	1	1.6%
100+ persons	1	1.6%
Total	64	100.0%

The most popular application types among respondents were web applications (76.2%) and cloud applications (68.3%). Other category contained seven answers: “Analytics”, “Automotive software”, “Integrations”, “Internal services without UI console (rest apis)”, “Test automation development”, “databases, database proxy” and “multidatabase systems”. (Table 8)

Table 8: Types of application / system development (N = 63)

Application / system type	N	%
Web applications	48	76.2%
Cloud applications	43	68.3%
Mobile applications	30	47.6%
Embedded systems	20	31.8%
Desktop applications	7	11.1%
Other	24	29.6%

* Single respondent could make multiple selections

Multiple respondents stated that they developed many types of applications. Using total was not applicable here.

4.2 DevOps usage and opinions

Based on the results, DevOps is well taken into use in Finland and practitioners mostly agree with the claims of DevOps concept, except shifting responsibilities to development

from operations. In organizations development team is gaining more responsibility and DevOps is implemented mostly on organization level.

Responses indicate that DevOps is mostly used among professionals in Finland. 76.5% of respondents reported that DevOps is used or implemented in their organization, 11.1% are evaluating it and 12.4% are not using it. (Table 9)

Table 9: DevOps usage in organizations

Usage	N	%
Yes	62	76.5%
No	10	12.4%
Evaluating it	9	11.1%
Total	81	100.0%

The reasons for not implementing include lack of knowledge or expertise about DevOps (50%) and lack of resources e.g. for acquiring new tools and expertise (50%). Other field contains also important answers such as *"Lack of access to cloud-based services"*, *"No need in our operations, but increasingly used among our customers"*, *"Small teams, fragmented technology and tools. Long release cycle is accepted."*. (Table 10)

Table 10: Reasons for not implementing DevOps (N= 10)

Usage	N	%
Lack of knowledge or expertise about DevOps	5	50%
Lack of resources e.g. for acquiring new tools and expertise	5	50%
Fragmented processes	2	20%
Lack of management support	2	20%
Cultural inhibitors	1	10%
Lack of proper monitoring tools	1	10%
Other	4	40%

Among the respondents that are using or evaluating DevOps (N = 70), DevOps is mostly implemented at organization level (38.6%), followed by in specific projects (28.6%). When looking at the effect of organization size to DevOps implementation level, small(41%) and medium(46%) organizations prefer organization level as their level of implementation and large organizations favor implementation in specific projects(37%). (Table 11)

Table 11: Implementation levels against organization sizes

Organization size	Implementation level			
	In specific projects	At team level	At product level	At organization level
Small	18.5%	25.9%	14.8%	40.7%
Medium	33.3%	12.5%	8.3%	45.8%
Large	36.8%	10.5%	26.3%	26.3%

The most agreed claims of DevOps concept are increasing automation (Mdn=5, IQR=0) and speeding up the delivery (Mdn=5, IQR=1). Shifting responsibilities from operations to development divided opinions the most (Mdn=3, IQR=2). (Table 12)

Table 12: Meaning of DevOps concept to respondents (N = 81)

Claim	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Mdn	IQR
Increasing automation in software process	1.2%	2.5%	3.7%	21.0%	71.6%	5	0
Shifting responsibilities to development from operations	7.4%	21.0%	23.5%	30.9%	17.3%	3	2
Speeding up the delivery of software changes without any breaks in system operations	-	1.2%	2.5%	40.7%	55.6%	5	1
Cross-functional collaboration within and between teams especially software development and operations	-	1.2%	6.2%	37.0%	55.6%	5	1

Most important practices to implement among respondents who had implemented or were evaluating DevOps in their respective organizations (N = 70) were automation and tool support for test, deployment and infrastructure (Mdn=5, IQR=1) and monitoring of software application by development, including customer usage behaviour (Mdn=4, IQR=1). Least favorable was ”software architectural changes to microservice system architecture” (Mdn=4, IQR=2.5). (Table 13)

Table 13: DevOps practices (to be) implemented as part of DevOps initiative (N = 70)

Practice	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
Automation and tool support for test, deployment, infrastructure	-	-	2.86%	22.86%	74.29%	5	1
Monitoring of software application by development, including customer usage behavior	1.43%	8.57%	8.57%	45.71%	35.71%	4	1
Software architectural changes to microservice system architecture	1.43%	15.71%	21.43%	38.57%	22.86%	4	2.5
Collaboration and team reorganization between development and operations	-	4.29%	21.43%	37.14%	37.14%	4	0.5

Support from customer, management and teams (Mdn=5, IQR=1) was seen as the most important factor contributing to successful DevOps implementation followed by clear understanding of what DevOps means to our organization including identification of concrete improvement areas (Mdn=4, IQR=1) and usage of agile practices/ideology (Mdn=4, IQR=1.5). (Table 14)

Table 14: Factors contributing to successful DevOps implementation (N = 69)

Factor	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
Trialing DevOps initiative to selected one or a small group of pilot applications	-	7.3	20.3%	46.4%	26.1%	4	1
Availability of supporting technology e.g. Docker, cloud computing in embedded systems	1.5%	4.4%	17.4%	36.2%	40.6%	4	2.5
Clear understanding of what DevOps means to our organization including identification of concrete improvement areas	-	1.5%	8.7%	37.7%	52.2%	4	1
Support from customer, management and teams	-	2.9%	11.6%	37.7%	47.8%	5	1
Usage of agile practices / ideology	-	2.9%	15.9%	37.7%	43.5%	4	1.5

4.3 Benefits and challenges of DevOps

Benefits and challenges featured in previous literature were featured in two questions which contained seven claims per question.

Two most important benefits were ”improves system quality” (Mdn=5, IQR=1) followed by ”improves release cycle time” (Mdn=4, IQR=1) and the least agreed upon is ”improves decision making in product development” (Mdn=4, IQR=1.5). (Table 15)

Table 15: Benefits of DevOps (N = 81)

Benefit	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
Improves release cycle time	1.2%	-	2.5%	38.2%	58.0%	4	1
Improves system quality	-	-	8.64%	38.3%	53.1%	5	1
Improves traceability of system changes	-	2.5%	23.5%	35.8%	38.3%	4	1
Improves organisational collaboration	-	1.2%	29.6%	33.3%	35.8%	4	2
Improves employee engagement	-	2.5%	24.7%	43.2%	29.6%	4	1.5
Improves decision making in product development	-	8.6%	30.9%	32.1%	28.4%	4	1.5
Improves efficiency of operations tasks in development	-	-	16.1%	40.7%	43.2%	5	1.5

DevOps challenges divide respondents' opinions more compared to benefits. The most agreed challenge is lack of common understanding for DevOps concept (Mdn=4, IQR=1.5) and difficulties in automating system testing (Mdn=4, IQR=1.5). Limitations imposed by application domain is the least problematic challenge (Mdn=3, IQR=2). (Table 16)

Table 16: Challenges of DevOps (N = 81)

Challenge	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
Lack of common understanding for DevOps concept	1.3%	5.0%	21.3%	53.8%	18.8%	4	1.5
High demand for skills and expertise on DevOps	-	9.9%	25.9%	39.5%	24.7%	4	2
Limitations imposed by legacy systems	0%	17.3%	18.5%	35.8%	28.4%	4	2
Limitations imposed by heterogeneous environments	1.2%	19.8%	25.9%	35.8%	17.3%	4	1.5
Limitations imposed by application domain	3.7%	27.2%	34.6%	28.4%	6.2%	3	2.5
Difficulties in automating system testing	8.6%	21.0%	17.3%	44.4%	8.6%	4	1.5
Difficulties in managing database schema upgrades in rapid and continuous releases	7.4%	22.2%	30.9%	30.9%	8.6%	3	2

4.4 DevOps Implementation

In addition to questions on team size and application type, technical details on software practices including continuous integration and deployment practices, release cycle times and automation levels of development chain were placed in technical section.

Multiple release cycle times are popular in responses. ”Not defined” (22%) and ”Monthly” (22%) releases were the most popular options followed by ”Daily” (16%) and ”Weekly” (16%) releases. ”Other” category contained the following responses: ”2 times / month”, ”2 weeks”, ”500 releases / month”, ”I work with seven teams with very different products and releases”, ”Installations are available daily, customers adopt them once in three years” and ”two weeks”. (Table 17)

Table 17: Release cycle times of sofware changes to production

Release cycle	N	%
Not defined	14	21.9%
Daily	10	15.6%
Weekly	10	15.6%
Monthly	14	21.9%
Several months to one year	8	12.5%
Not in production	2	3.1%
Other	6	7.4%
Total	64	100.0%

Commit frequency and reasons for less frequent commits were combined using two questions. More frequent than daily commits were most popular option (52%) followed by daily commits (30%). 19% of respondents reported less frequent than daily (19%) commits. (Table 18)

Table 18: Commit frequency to mainline by developers

Release	N	%
More frequently	33	51.6%
Daily	19	29.7%
Less frequently than daily	12	18.8%
Total	64	100.0%

The respondents that had less frequent than daily commits were asked to specify how they agree with different reasons for less frequent commits from three options. Most agreed was ”It is hard to split certain feature to smaller pieces” followed by ”Delivery process is time-consuming or it is too complicated to deliver changes”. The respondents were mostly neutral regarding the evident value when delivering changes. (Table 19)

Table 19: Reasons for less frequent commits to mainline (N = 12)

Reason	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
It is hard to split certain feature to smaller pieces	8.3%	-	16.7%	58.3%	16.7%	4	1
Delivery process is time-consuming or it is too complicated to deliver changes	-	33.3%	8.3%	50.0%	8.3%	4	2
There is no evident value when delivering changes	-	25.0%	41.7%	25.0%	8.3%	3	2

Automation levels of releasing code into test environment was more common than to production environment. 80% of respondents reported having an automated chain into test environment and 31% had automated chain to production environment. (Table 20)

Table 20: Automated chain from code commit to test and production environments

Automated chain to...	N	%
Test environment		
Yes	51	79.7%
No	13	20.3%
Production environment		
Yes	20	31.3%
No	44	68.7%
Total	64	100.0%

Continuous integration was featured in multiple technical questions. First of these is: ”How effectively is continuous integration used in projects in your organization?”. Most of the respondents reported that it was used in most (41%) or all (38%) projects. 6% of respondents reported that it was not used at all. (Table 21)

Table 21: Usage of Continuous Integration in projects

Usage	N	%
In all projects	24	37.5%
In most projects	26	40.6%
Only in few projects	10	15.6%
Not at all	4	6.3%
Total	64	100.0%

Three statements of manual and automated procedures to outputs of CI for production deployment were queried to find out deployment procedures before the actual deployment. Instant creation of deployable package (Mdn=4, IQR=2.5) and static and dynamic code analysis of the system (Mdn=4, IQR=2.5) were agreed with even though there was some dispersion. Feature toggles (Mdn=3, IQR=1.5) weren’t as popular as these two. (Table 22)

Table 22: Interventions / actions to CI outputs for production deployment (N = 64)

Action	SD (1)	D (2)	N (3)	A (4)	SA (5)	M	IQR
Feature toggles are used for unready features	10.9%	17.2%	31.3%	31.3%	9.4%	3	1.5
Static and dynamic code analysis is applied	12.5%	17.2%	4.7%	43.8%	21.9%	4	2.5
Deployable package is immediately created	4.7%	17.2%	12.5%	37.5%	28.1%	4	2.5

Seven claims of CI best practices were rated by respondents to discover the most important practices of CI in this context. Two most positively viewed practices are code review by other developers (Mdn=4, IQR=0.5) and immediate communication of CI status to other team members (Mdn=4, IQR=1.5). (Table 23)

Table 23: CI best practices in software development process (N = 64)

Practice	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
Daily commits to mainline	4.7%	4.7%	20.3%	34.4%	35.9%	3	2
New code is reviewed by other developers	1.6%	3.1%	9.4%	35.9%	50.0%	4	0.5
Successful CI build presumes that all automated tests have passed	3.1%	4.7%	12.5%	26.6%	53.1%	5	1
A significant portion of all testing activities are automated	7.8%	9.4%	4.7%	32.8%	45.3%	4	1
Deployment to production is trivial and can be performed at any given moment	7.8%	10.9%	9.4%	28.1%	43.8%	4	2.5
Immediate communication of CI status to team members (N = 63)	3.2%	3.2%	7.9%	44.4%	41.3%	4	1.5
Fixing detected faults is top priority	-	6.3%	10.9%	39.1%	43.8%	4	0.5

Four statements on benefits and three statements on challenges of CI were requested to grade for the next part. Developer productivity (Mdn=4, IQR=1) was seen as the biggest benefit among the options, followed by improvements on project predictability (Mdn=4, IQR=1.5). Most agreed challenge is the build process of large multi-platform software systems and the difficulty of understanding and maintaining them (Mdn=4, IQR=1) followed by build process taking unreasonable time to complete(Mdn=4, IQR=1). Other challenges were mostly disagreed with. (Table 24)

Table 24: Benefits and challenges of Continuous Integration (N = 64)

Statement	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
Benefits							
Improves communication between developers	-	4.7%	18.8%	48.4%	28.1%	4	1
Improves developer productivity	-	3.1%	9.4%	42.2%	45.3%	4	1
Improves project predictability (N = 63)	-	1.6%	17.5%	46.0%	34.9%	4	1.5
Improves developers' ability to effectively deal with rebasing and merging	1.6%	1.6%	26.6%	42.2%	28.1%	3	1
Challenges							
Increase in development overhead due to numerous merge conflicts and rebasing efforts	21.9%	39.1%	26.6%	12.5%	-	3	1.5
Build process of large, multi-platform software systems are difficult to understand and maintain (N = 63)	3.2%	19.1%	23.8%	41.3%	12.7%	4	1
Build process takes an unreasonable long time to complete	10.9%	29.7%	26.6%	25.0%	7.8%	4	1

The deployment of software were featured in two sets of claims: best practices and challenges of deployment. Best practices according to respondents are using the same deliverable from CI to deploy to test, acceptance and production environments (Mdn=5, IQR=1), version control of configuration scripts for infrastructure provisioning and management (Mdn=5, IQR=1) and rollback of made changes (Mdn=4, IQR=1.5). (Table 25)

Table 25: Best practices of software deployment (N = 63)

Statement	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
Same deliverable from CI is deployed to test, acceptance and production environments	-	4.8%	6.4%	46.0%	42.9%	5	1
Development and test environments mirror or clone the production environment	-	14.3%	12.7%	49.2%	23.8%	4	2
Deployed software changes can be rolled back whenever necessary	-	6.4%	9.3%	46.0%	38.1%	4	1.5
Configuration scripts for infrastructure provisioning and management are version controlled	-	6.4%	7.9%	27.0%	58.7%	5	1
Deployment process is fully automated (N = 62)	4.8%	6.5%	11.3%	41.9%	35.5%	5	1
Development team continuously monitor software during and after deployment of software updates	-	9.5%	14.3%	31.8%	44.4%	4	1.5

Five statements were given to respondents to measure challenges of deployment process. The most agreed challenges according to respondents were difficulty of testing dependency specifications (Mdn=4, IQR=1.5) and difficulty of rolling back software updates (Mdn=3, IQR=2). Overall the responses were distributed heavily and many respondents disagreed with ”Deployment actions are only performed by operations, not by development team” (Mdn=2, IQR=2.5) and ”Limited technology support to fully automate deployment of software updates” (Mdn=3, IQR=2). (Table 26)

Table 26: Challenges of software deployment process (N = 62)

Statement	SD (1)	D (2)	N (3)	A (4)	SA (5)	Mdn	IQR
It is difficult to test whether dependency specifications are complete, except at runtime	-	25.8%	37.1%	30.7%	6.5%	4	1.5
It is difficult to roll back software updates	4.8%	32.3%	22.6%	32.7%	8.1%	3	2
Deployment actions are only performed by operations, not by development team	21.0%	32.3%	22.6%	19.4%	4.8%	2	2.5
Deployment in heterogeneous environment is at large done manually based on deployment documentation (N = 59)	15.3%	23.7%	28.8%	27.1%	5.1%	3	2
Limited technology support to fully automate deployment of software updates	21.0%	29.0%	17.7%	24.2%	8.1%	3	2

4.5 Tooling

The final part of the questionnaire and the technical section was focused on tools used and the respondents could select multiple for each category and specify other tools used in the ”other” category. The respondents reported usage of multitude of tools. The most popular tools among respondents were Jenkins(73.0%) in CI, Kibana(46.2%) in monitoring, Amazon AWS(61.4%) in cloud, Custom / Inhouse scripts(22.5%) in deployment and Ansible(58.5%) in configuration and provisioning. ”Other” categories in deployment and monitoring were very large with different combinations of tools that were not present in the options. (Table 27)

Table 27: DevOps tools usage

Tool	Usage	Tool	Usage
Continuous Integration (N = 63)		**Deployment (N = 40)**	
Jenkins	73.0%	Custom / Inhouse scripts	22.5%
Travis CI	17.5%	Deployment manager	12.5%
GitLab CI	11.1%	Capistrano	2.5%
Bamboo	9.5%	SmartFrog	2.5%
CircleCI	8.0%	Other	62.5%
VSTS / TFS	6.2%		
Other	14.3%		
Monitoring (N = 52)		**Config / Provisioning (N = 53)**	
Kibana	46.2%	Ansible	58.5%
New Relic	32.7%	Chef	18.9%
Grafana	11.5%	TerraForm	18.5%
AWS cloudwatch	9.6%	Puppet	15.1%
Nagios	7.7%	SaltStack	7.4%
Dynatrace	1.9%	VSTS	3.8%
Other	50.0%	Other	22.6%
Cloud /IaaS / PaaS (N = 57)			
Amazon AWS	61.4%		
Azure	31.6%		
Google Cloud	22.8%		
Other	29.8%		

*Multiple selections possible in each category

The tools are usually featured in industry reports supported by tool vendors like Atlassian (The Scoop on Technical Support & Development, February 2016) or Puppet (State of DevOps). The different categories of tools have also been featured in a previous study (Vaasanthi, Kumari, & Kingston, 2017).

5 Discussion

The aim of this study was to find out whether DevOps is used or not, how is it used and what type of benefits, challenges and practices are related to the use of DevOps in Finnish software industry. Underlying practices such as continuous integration and deployment along with the tools used were featured in the technical section of the questionnaire.

According to the data, DevOps was mostly used in respondents' respective companies. It seemed that overall professionals are aligned with the claims and statements on DevOps extracted from the literature. Shifting responsibilities from operations to developers was polarized but at the same time in deployment challenges respondents disagreed with deployment actions being performed only by operations. Mixing responsibilities was deemed problematic by other parties due to increased learning and issues with configuration management (Nybom et al., 2016). Microservice architecture also divided opinions even though in literature microservices was reported important in realizing the full potential of DevOps (Balalaie et al., 2016). Implementing this type of architecture probably requires a lot of refactoring or can be impossible due to underlying requirements for industries, services and especially embedded software usage. For example the following related response was left by one respondent in free comments field:

> Our environment contains legacy from 80-90's to just released internal and external/public software / applications. Thus environment is heterogeneous and extremely complicated. **Also in many places we are not allowed to update software automatically/seamlessly even if technically possible**. We have agreements that require us to notify our customers several weeks in advantage before starting major updates so that they can prepare for possible outages and ramp up their alternative/backup systems. So some of the stuff are "mission critical".

Aspects like meaning of DevOps and factors contributing to successful DevOps adoption were directly asked from the respondents. Automation was regarded highly as the most agreed concept and practice of DevOps but in contrast, in the comments one of the respondents left a contrasting comment: "DevOps should be about people and their interactions, not purely about automation...". Many previous articles also regard automation as core principle or practice of DevOps (de França et al., 2016; Jabbari et al., 2016; Gupta et al., 2017; Ebert et al., 2016).

Continuous integration was seen as beneficial in improving developer productivity but challenging in build processes of large, multiplatform systems. Jenkins was used by most of the respondents to help in the process. For CI outputs, code analysis and deployable packages are created in most cases, feature toggles used a bit less. Best practices in the CI process were all similarly agreed by respondents. CI is featured in several other studies as well as a building block of DevOps and as its own category of tools or questions (Jabbari et al., 2016; Bartusevics, 2017; Boettiger, 2015).

Deployment process practices were also agreed but full automation and mirroring production environment received some disagreements. Opinions on deployment challenges were much more divided which would suggest that some of the respondents were facing issues in this field. Tools used in deployment were mostly customized or one-of-a-kind

in-house built scripts. It could be that the deployment process in these problematic cases was somehow more complicated than just putting software on an owned server as one of the comments suggested: " ... Production operating environment also dictates some challenges, e.g. some deployments are done to on-premise setting where all tools/processes or any DevOps related matter is in order but the customer has its own processes to run deployments."

Benefits and challenges of DevOps were summarized using two likert-scale questions. Among benefits decision making in product development was slightly less agreed than others. The most agreed upon benefits – Improving quality and improving release cycle time – were in line with other research on DevOps focusing on concerns of quality (Di Nitto et al., 2016) and as a part of process in the DevOps maturity model proposed (Bucena & Kirikova, 2017). Quality was also one dimension of factors in previous survey study on Agile (Stankovic et al., 2013). Faster releases were also featured in multiple studies as they are one of the main benefits of DevOps and automation (Balalaie et al., 2016; Shahin, Babar, et al., 2017; Airaj, 2017).

Main challenge from the provided options was the common understanding for DevOps concept. This was also highlighted in several previous academic and industrial studies on DevOps (Jabbari et al., 2016; Virmani, 2015; de França et al., 2016; Brown et al., 2016). Even though there are certain differences in environments, the main goals of DevOps should be well communicated and explained to achieve most of it. As one of the respondents commented on DevOps: "DevOps should be about people and their interactions, not purely about automation. ... DevOps means good, but the name makes it do wrong things."

6 Conclusion

This thesis included an overview of DevOps, consisting of benefits, challenges, practices and meaning of the consept. These aspects were formed into questions in a questionnaire which was presented to the software industry professionals inside Finland. The objective of this study was to describe the perception and application of DevOps among ICT practitioners in Finland. The results are based on survey data collected from 81 Finnish respondents in 2018.

According to the results and answering the perception of adoption and use (**RQ1**), DevOps was widely taken into practice in Finland and practiced in smaller (<10 members) teams. All types of applications had their fair share of users, Web and cloud applications were the most popular domains. Automation was the most important meaning of the concept and also most agreed practice. Clear understanding was regarded as the most important factor contributing to DevOps adoption.

Regarding the engineering practices and tools (**RQ2**), CI was used in almost all organizations with automated chain to test environment but not to production environment. Code review was regarded as the best practice of CI. Developer productivity was a popular benefit and build process of large systems was deemed challenging. Most common tool used in CI process was Jenkins, in monitoring Kibana, in cloud Amazon AWS, in deployment Other and in configuration Ansible.

The final question focused on benefits and challneges perceived (**RQ3**). Most important benefits were system quality and increasing release cycle time and lack of common understanding was most agreed challenge.

The main contributions of this study are: the results, the questionnaire and feedback. The collected results provide insight on the differences of current research and practice on DevOps in Finland. The questionnaire instrument created can be used again and improved according to the needs of the users. Open comments on the study are meaningful and could be used in conducting a better and more focused questionnaire using a bigger or otherwise different target group.

6.1 Limitations and threats to validity

Main limitation of the questionnaire was low amount of respondents. Even though the questionnaire was advertised in multiple groups and channels in Slack, it didn't get over 100 responses. Garousi et al. had also a fairly small amount of responses (46) while targeting Turkish IT companies (Garousi et al., 2015). This could have been due to the DevOps orientation of the questionnaire which could have already made answering impossible if the respondent didn't know anything about DevOps which would have been one type of a result.

The selected sharing methods didn't allow calculating response rates because using open invitations in Slack and LinkedIn make calculating those impossible. Sharing the survey using Slack and LinkedIn might have subjected the questionnaire to answers from respondents outside of target group and also the threat of corrupted answers.

Using mainly quantitative data is a considerable limitation even though it was crucial to keep the answer time low enough for respondents to finish it. Length of the survey limited the amount of data collected and some interesting results were possibly missed because of this. Qualitative questions could have given more information behind the Likert scale answers.

In addition to limitations, there are four threats to validity. First, the convenient selection of potential participants by utilizing e-mails, e-mail lists and slack groups could have an effect on the internal validity of the questionnaire. Multiple different groups were selected in order to mitigate this threat. Secondly, the possibility of drop outs in the middle of the questionnaire could affect the reliability of the data. For this reason the questionnaire was piloted multiple times to make sure that the length and time are not too demanding and there are skip rules in place should the respondent want to skip sections. Thirdly, the external validity could be threatened by the small sample size however the background of the respondents indicates that there are multiple different respondents. Fourth and final threat is the reliability of Likert scale. The distance between options that indicate agreement and strong agreement might be different among different respondents. This is a common issue in Likert scale and the results are not presented as absolute answers due to potential threats in the scale itself.

6.2 Future research

Further research in the subject is required and the instrument created for this study could be used in another survey in different context. Questionnaire could also be developed further to include more qualitative questions which in this case were cut due to length issues found during piloting. Since the data was split to DevOps and CI/CD proficiencies, it could be focused entirely to one of these and to include open ended questions to gain deeper insight on issues behind the opinions.

Interesting facts were discovered from the open-ended final comments which suggests that conducting qualitative research such as a more open survey or an interview study would probably generate interesting results in the same context. One of the respondents was already offering an interview for this study within the questionnaire comments. For longer questionnaires, different techniques such as different incentives for completing the survey could be considered to keep respondent interest high.

In future studies, the survey instrument could include questions on the reasoning behind the measured opinions and practices of DevOps. Furthermore, a study is needed to investigate the effects of other factors, such as the background of respondents, have on the perception and experiences regarding the DevOps concept, practices and impacts. Another interesting aspect would be to investigate usage of DevOps tools in different contexts. Also, an option that indicates no previous knowledge of DevOps could be valuable.

The questionnaire could also be used in a single company to determine where the problem points are during DevOps implementation or to get insight on how software personnel are viewing different aspects of DevOps.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Airaj, M. (2017). Enable cloud DevOps approach for industry and higher education. *Concurrency Computation*, *29*(5), 1–11. doi: 10.1002/cpe.3937

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

Bartusevics, A. (2017). Automation of continuous services: what companies of Latvia says about it? *Procedia Computer Science*, *104*, 81–88.

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

Boettiger, C. (2015). An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, *49*(1), 71–79.

Brown, A., Forsgren, N., Humble, J., Kersten, N., & Kim, G. (2016). 2016 state of devops report. *Puppet labs*.

Bruneo, D., Longo, F., Merlino, G., Peditto, N., Romeo, C., Verboso, F., & Puliafito, A. (2015). Enabling collaborative development in an openstack testbed: The cloudwave use case. In *Proceedings of the seventh international workshop on principles of engineering service-oriented and cloud systems* (pp. 24–30). Piscataway, NJ: IEEE Press.

Bucena, I., & Kirikova, M. (2017). Simplifying the devops adoption process. In J. Björn (Ed.), *Pre-bir forum, bir workshops and doctoral consortium 2017* (Vol. 1898). Aachen. Retrieved from `http://ceur-ws.org/Vol-1898`

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

Cito, J., Leitner, P., Fritz, T., & Gall, H. C. (2015). The making of cloud applications: An empirical study on software development for the cloud. In *Proceedings of the 10th joint meeting on foundations of software engineering* (pp. 393–403). New York: ACM Press. doi: 10.1145/2786805.2786826

Debois, P. (2008). *Agile 2008 Toronto: Agile Infrastructure and Operations Presentation.* Retrieved 2018-01-23, from `http://www.jedi.be/blog/2008/10/09/agile-2008-toronto-agile-infrastructure-and-operations-presentation/`

de França, B. B. N., Jeronimo, H., & Travassos, G. H. (2016). Characterizing DevOps by Hearing Multiple Voices. *Proceedings of the 30th Brazilian Symposium on Software Engineering - SBES '16*, 53–62. doi: 10.1145/2973839.2973845

DevOps Finland. (n.d.). Retrieved 2018-04-27, from `http://www.devopsfinland.org/`

Di Nitto, E., Jamshidi, P., Guerriero, M., Spais, I., & Tamburri, D. A. (2016). A software architecture framework for quality-aware DevOps. In *2nd international workshop on quality-aware devops* (pp. 12–17). ACM Press. doi: 10.1145/2945408.2945411

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). *A decade of agile methodologies: Towards explaining agile software development.* Elsevier.

Dyck, A., Penners, R., & Lichter, H. (2015). Towards definitions for release engineering and DevOps. *Proceedings - 3rd International Workshop on Release Engineering, RELENG 2015*, 3–3. doi: 10.1109/RELENG.2015.10

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

Garousi, V., Coşkunçay, A., Betin-Can, A., & Demirörs, O. (2015). A survey of software

engineering practices in Turkey. *Journal of Systems and Software*, *108*, 148–177. doi: 10.1016/j.jss.2015.06.036

Gupta, V., Kapur, P. K., & Kumar, D. (2017). Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling. *Information and Software Technology*, *92*, 75–91. doi: 10.1016/j.infsof.2017.07.010

Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). What is DevOps?: A systematic mapping study on definitions and practice. *Proceedings of the Scientific Workshop Proceedings of XP2016*, 1–11. doi: 10.1145/2962695.2962707

Jones, S., Noppen, J., & Lettice, F. (2016). Management challenges for DevOps adoption within UK SMEs. In *2nd international workshop on quality-aware devops - qudos 2016* (pp. 7–11). New York, NY: ACM Press. doi: 10.1145/2945408.2945410

Kamuto, M. B., & Langerman, J. J. (2017). Factors inhibiting the adoption of DevOps in large organisations: South African context. In *2017 2nd IEEE international conference on recent trends in electronics, information communication technology (RTEICT)* (p. 48-51). Piscataway, NJ: IEEE. doi: 10.1109/RTEICT.2017.8256556

Koodiklinikka. (n.d.). Retrieved 2018-04-27, from `https://koodiklinikka.fi/`

Leppänen, M., Kilamo, T., & Mikkonen, T. (2015). Towards post-agile development practices through productized development infrastructure. In *Proceedings of the second international workshop on rapid continuous software engineering* (pp. 34–40). Piscataway, NJ: IEEE Press.

Leppänen, M., Makinen, S., Pagels, M., Eloranta, V.-P., Itkonen, J., Mäntylä, M., & Männistö, T. (2015, 3). The highways and country roads to continuous deployment. *IEEE Software*, *32*, 64-72. doi: 10.1109/MS.2015.50

Lindgren, E., & Münch, J. (2016). Raising the odds of success: the current state of experimentation in product development. *Information and Software Technology*, *77*, 80 - 91. doi: https://doi.org/10.1016/j.infsof.2016.04.008

Lwakatare, L. E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H. H., Bosch, J., & Oivo, M. (2016). Towards DevOps in the embedded systems domain: Why is it so hard? In *Proceedings of the 49th hawaii international conference on system sciences (hicss)* (pp. 5437–5446). Piscataway, NJ: IEEE Press. doi: 10.1109/HICSS.2016.671

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of devops. In C. Lassenius, T. Dingsøyr, & M. Paasivaara (Eds.), *Agile processes in software engineering and extreme programming* (pp. 212–217). Cham: Springer International Publishing.

Mäkinen, S., Leppänen, M., Kilamo, T., Mattila, A.-L., Laukkanen, E., Pagels, M., & Männistö, T. (2016). Improving the delivery cycle: A multiple-case study of the toolchains in finnish software intensive enterprises. *Information and Software Technology*, *80*, 175–194. doi: 10.1016/j.infsof.2016.09.001

Nybom, K., Smeds, J., & Porres, I. (2016). On the impact of mixing responsibilities between Devs and Ops. In H. Sharp & T. Hall (Eds.), *Agile processes, in software engineering, and extreme programming* (pp. 131–143). Cham: Springer International Publishing.

O'leary, Z. (2004). *The essential guide to doing research.* Sage.

Rajkumar, M., Pole, A. K., Adige, V. S., & Mahanta, P. (2016). DevOps culture and its impact on cloud delivery and software development. In *International Conference on Advances in Computing, Communication, Automation (ICACCA).* Piscataway, NJ: IEEE Press. doi: 10.1109/ICACCA.2016.7578902

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). Devops adoption benefits and challenges in practice: a case study. In *International Conference on Product-Focused Software Process Improvement* (pp. 590–597).

Rodríguez, P., Markkula, J., Oivo, M., & Turula, K. (2012). Survey on agile and lean us-

age in finnish software industry. In *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 139–148). Piscataway, NJ: IEEE Press.

Shahin, M., Babar, M. A., Zahedi, M., & Zhu, L. (2017). Beyond continuous delivery: an empirical investigation of continuous deployment challenges. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 111–120). Piscataway, NJ.

Shahin, M., Zahedi, M., Babar, M. A., & Zhu, L. (2017). Adopting continuous delivery and deployment: Impacts on team structures, collaboration and responsibilities. In *Proceedings of the 21st international conference on evaluation and assessment in software engineering* (pp. 384–393). New York: ACM Press. doi: 10.1145/3084226.3084263

Smeds, J., Nybom, K., & Porres, I. (2015). DevOps: A Definition and Perceived Adoption Impediments. In C. Lassenius, T. Dingsøyr, & M. Paasivaara (Eds.), *Agile processes in software engineering and extreme programming* (pp. 166–177). Cham: Springer International Publishing.

Soni, M. (2015). End to end automation on cloud with build pipeline: the case for DevOps in insurance industry, continuous integration, continuous testing, and continuous delivery. In *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)* (pp. 85–89). Piscataway, NJ: IEEE.

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

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, *86*(6), 1663–1678. doi: 10.1016/j.jss.2013.02.027

Straub, D. W., Gefen, D., & Boudreau, M.-C. (2005). *Quantitative research* (Vol. 1). Amsterdam: Elsevier.

Vaasanthi, R., Kumari, V. P., & Kingston, S. P. (2017). Analysis of devops tools using the traditional data mining techniques. *International Journal of Computer Applications*, *161*(11).

Van Der Linden, F., Bosch, J., Kamsties, E., Känsälä, K., & Obbink, H. (2004). Software Product Family Evaluation. *Software Product Lines*, 110–129. doi: 10.1007/b100081

Virmani, M. (2015). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In *Fifth International Conference on the Innovative Computing Technology (INTECH 2015)* (pp. 78–82). Piscataway, NJ: IEEE Press.

A Survey instrument

The final questionnaire contained total of 30 questions in two pages with two rules in place to focus on different aspects. All of the questions were designed to be quantitative and two rules were created to enable more relevant insights on different respondents. First rule was tied to the question 5: “Is DevOps used or implemented in your organisation?”. If the answer was ”Yes” or ”We are evaluating it”, questions 6-8 were shown. If ”No” then question 9 was shown. The second rule was related to question 12: “Can you answer more technical questions?”. Answering ”No” ended the questionnaire there.

The following list contains the final survey instrument with the final questions used in the Webropol survey. Options marked with (+) or (-) symbols indicate changes in the following questions.

1. What is the approximate size of your organization in Finland?
   (a) 1 – 50 employees
   (b) 51 – 100 employees
   (c) 101 – 500 employees
   (d) 501 – 1000 employees
   (e) 1000+ employees
2. What is your current main working role?
   (a) Software developer / Software engineer
   (b) Software architect
   (c) Project manager
   (d) Quality assurance / Test engineer
   (e) Operations staff / System administrator
   (f) Other
3. How are development and operations organised in your organization?
   (a) Separate development and operations teams with high collaboration
   (b) Separate development and operations teams with a facilitator (an individual or team) in between them
   (c) Development team gaining more responsibility for operations, and operations existing in a small team
   (d) No visible operations team
4. How would you agree or disagree with the claims below about what the concept of DevOps means to you? (1 – 5)
   (a) Increasing automation in software process
   (b) Shifting responsibilities to development from operations
   (c) Speeding up the delivery of software changes without any breaks in system operations

   (d) Cross-functional collaboration within and between teams especially software development and operations

5. Is DevOps used or implemented in your organisation?

   (a) Yes (+)
   (b) No (-)
   (c) We are evaluating it (+)

6. At what level have you applied DevOps? (+)

   (a) In specific projects
   (b) At team level
   (c) At product level
   (d) At organization level

7. What specific practices you have / are seeking to implement as part of DevOps initiative? (+) (1 – 5)

   (a) Automation and tool support for test, deployment, infrastructure
   (b) Monitoring of software application by development, including customer usage behaviour
   (c) Software architectural changes to microservice system architecture
   (d) Collaboration and team reorganization between development and operations

8. How would you agree or disagree with the following as factors that contribute/will contribute to successful DevOps implementation? (+) (1 – 5)

   (a) Trialling DevOps initiative to selected one or a small group of pilot applications
   (b) Availability of supporting technology e.g., Docker, cloud computing in embedded systems
   (c) Clear understanding of what DevOps means to our organization including identification of concrete improvement areas
   (d) Support from customer, management and teams
   (e) Usage of agile practices / ideology

9. Why has DevOps not been applied to your team? (-) (Select multiple)

   (a) Cultural inhibitors
   (b) Fragmented processes
   (c) Lack of management support
   (d) Lack of resources e.g., for acquiring new tools and expertise
   (e) Weak tool support
   (f) Lack of support for security and regulatory compliance
   (g) Misalignment of tools between software development and operations activities

(h) Lack of proper monitoring tools

(i) Lack of knowledge / expertise about DevOps

(j) Other

10. How would you agree or disagree with the following benefits of applying DevOps in practice? (1 – 5)

(a) Improves release cycle time

(b) Improves system quality

(c) Improves traceability of system changes

(d) Improves organizational collaboration

(e) Improves employee engagement

(f) Improves decision making in product development

(g) Improves efficiency of operations tasks in development

11. How would you agree or disagree with the following challenges of Devops? (1 – 5)

(a) Lack of common understanding for DevOps concept

(b) High demand for skills and expertise on DevOps

(c) Limitations imposed by legacy systems

(d) Limitations imposed by heterogeneous environments

(e) Limitations imposed by application domain

(f) Difficulties in automating system testing

(g) Difficulties in managing database schema upgrades in rapid and continuous releases

12. Can you answer more technical questions?

(a) Yes

(b) No

13. What type of application/system development are you currently working on or involved with? (Select multiple)

(a) Cloud applications

(b) Web applications

(c) Embedded systems

(d) Mobile applications

(e) Desktop applications

(f) Other

14. What is the average size of the development team of the system you are currently working on?

(a) 1 – 5 persons

(b) 6 – 10 persons

(c) 11 – 20 persons

(d) 21 – 50 persons

(e) 51 – 100 persons

(f) 100+ persons

15. What is the release cycle time of software changes to production of the system you are currently working on/involved?

(a) Not defined

(b) Daily

(c) Weekly

(d) Monthly

(e) Several months to one year

(f) More than a year

(g) Not in production

(h) Other

16. Is there an automated chain from code commit to test / production environment?

17. How effectively is continuous integration used in projects in your organization?

(a) In all projects

(b) In most projects

(c) Only in few projects

(d) Not at all

18. What is the commit frequency to mainline by developers?

(a) More frequently than daily

(b) Daily

(c) Less frequently than daily (-)

19. How would you agree or disagree with the following statements when considering the reasons for less frequent commits to mainline? (-)

(a) It is hard to split certain feature to smaller pieces

(b) Delivery process is time-consuming or it is too complicated to deliver changes

(c) There is no evident value when delivering changes

20. How would you agree or disagree with the following interventions / actions to CI outputs for production deployment of the system you are currently working on? (1 – 5)

(a) Feature toggles are used for unready features

(b) Static and dynamic code analysis is applied (e.g. checking code test coverage)

(c) Deployable package (e.g., VM image) is immediately created

21. How would you agree or disagree with the following statements of CI best practices in your software development process?

    (a) Daily commits to mainline

    (b) New code is reviewed by other developer(s)

    (c) Successful CI build presumes that all automated tests have passed

    (d) A significant portion of all testing activities are automated

    (e) Deployment to production is trivial and can be performed at any given moment

    (f) Immediate communication of CI status to team members

    (g) Fixing detected faults is top priority

22. How would you agree or disagree with the following statements about benefits of continuous integration? (1 – 5)

    (a) Improves communication between developers

    (b) Improves developer productivity

    (c) Improves project predictability

    (d) Improves developers' ability to effectively deal with rebasing and merging

23. How would you agree or disagree with the following statements about challenges of continuous integration? (1 – 5)

    (a) Increase in development overhead due to numerous merge conflicts and rebasing efforts

    (b) Build process of large, multi-platform software systems are difficult to understand and maintain

    (c) Build process takes an unreasonable long time to complete

24. How would you agree or disagree with the following statements about best practices of deployment while taking into consideration infrastructure? (1 – 5)

    (a) Same deliverable from CI is deployed to test, acceptance and production environment

    (b) Development and test environments mirror or clone the production environment

    (c) Deployed software changes can be rolled back whenever necessary

    (d) Configuration scripts for infrastructure provisioning and management are version controlled

    (e) Deployment process is fully automated

    (f) Development team continuously monitor software during and after deployment of software updates

25. How would you agree or disagree with the following statements about challenges of deployment process?

    (a) It is difficult to test whether dependency specifications are complete, except at runtime

(b) It is difficult to roll back software updates

(c) Deployment actions are only performed by operations, not by development team

(d) Deployment in heterogeneous environment is at large done manually based on deployment documentation

(e) Limited technology support to fully automate deployment of software updates

26. Tools - Continuous Integration (select multiple)

(a) Jenkins

(b) Travis CI

(c) Bamboo

(d) Other... (specify)

27. Tools - Deployment (select multiple)

(a) Otto

(b) Deployment manager

(c) SmartFrog

(d) Other... (specify)

28. Tools - Monitoring (select multiple)

(a) Kibana

(b) New Relic

(c) Dynatrace

(d) Other... (specify)

29. Tools - Config / Provisioning (select multiple)

(a) Chef

(b) Puppet

(c) Ansible

(d) Other... (specify)

30. Tools - Cloud / Infrastructure as a Service / Platform as a Service (select multiple)

(a) Amazon Web Services (AWS)

(b) Azure

(c) Google Cloud

(d) Other... (specify)