MASTER

Demystifying NoOps
operational model, challenges and insights from the trenches

Șerban, Dragos Mihai

Award date:
2021

Department of Mathematics and Computer Science
Software Engineering and Technology Research Group

Demystifying NoOps: operational model, challenges and insights from the trenches

Master Thesis

Dragos Mihai Serban

Supervisors:
dr. Alexander Serebrenik
dr. Damian A. Tamburri

Committee:
dr. Odysseas Papapetrou

Version 1.0

Eindhoven, August 2021

Abstract

In recent years, the rise of DevOps solved many of the problems of early software development methodologies by improving the collaboration between development and operations teams. However, one of the main issues of DevOps is the assumption that developers want to be involved in operations-related activities, which is not always true. This issue led to the creation of NoOps - the promise that developers would focus solely on development activities, eliminating their involvement in any operational activities. Due to its novelty, NoOps is a concept that has not yet been formalized in peer-reviewed literature. This thesis aims at providing an overview of the NoOps concept by analyzing both practitioners' literature and online discussions. This overview focuses on the different dimensions of the NoOps definition, the positioning of NoOps in relation to similar concepts, the associated characteristics, and the reported advantages and challenges. To this end, we first conduct a systematic literature mapping study, performing qualitative analysis on online practitioners' literature. Then, we analyze NoOps-related online discussions using a combination of topic modeling and opinion mining, in order to account for practitioners' viewpoint. Finally, we perform a comparison between the findings emerging from the analysis of practitioners' literature and practitioners' viewpoint.

Preface

First and foremost, I would like to thank my supervisors, dr. Alexander Serebrenik and dr. Damian A. Tamburri for their excellent guidance, support and understanding throughout this graduation project. They have always been there whenever I had doubts and always gave me opportunities to improve my work and develop myself. I have learned a lot from every discussion we had and I am deeply grateful for the chance of being their student. I would also like to thank them for showing me how fun software engineering research can be.

I would also like to thank dr. Odysseas Papapetrou for accepting to be part of my graduation committee, but also for helping me discover the amazing field of distributed data processing.

Moreover, I would like to thank my girlfriend, my parents and my friends for always being there for me throughout my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the early days of software engineering, linear software development models were defined, such as the V-Model [26] or the Waterfall approach [76]. However, these models have been widely criticized for various reasons, including the lack of flexibility [77] and improper communication between the project teams involved in different development phases [72]. The drawbacks of these traditional models led to the conception of the Agile methodology, which advocates the iterative development of software products through self-organizing teams and close collaboration with the customer [21]. According to the 14th State of Agile Report [20], which surveyed 1121 Agile practitioners from all over the world, the most common reasons for adopting Agile within organizations are the acceleration of software delivery (71% of responses) , flexibility in changing requirements and priorities (63%) and increased productivity (51%). When it comes to the Agile adoption benefits, the report highlights adaptability to change (70%), reduced time-to-market (60%) and higher team morale (59%). Despite these benefits, Agile has one important drawback - although the software development timelines were shortened, the resulting products could not be delivered at the same speed, due to a lack of focus on the operational viewpoint [75].

Within organizations, software development and operations have often been separated into different business units, with different goals, processes and mindsets [61]. Although the goal of software development teams is to create and innovate, the operations teams focus on performance and reliability [61]. Hence, despite the fact that Agile improved the software development process, the misalignment and the lack of communication between development and operations units resulted in delivery bottlenecks [85].

The DevOps concept was conceived as a solution for the collaboration gap between software development and operations teams [42]. Fundamental to DevOps is a set of practices that aims at bringing the development and operations teams together, such that they can build, test and release software quicker [7]. The DevOps set of practices combines both technical and cultural elements. For instance, on a technical level, DevOps involves the use of practices such as Continuous Integration, Infrastructure as Code, as well as Automated Monitoring [27]. On a cultural level, DevOps encourages open collaboration, continuous learning and responsibility alignment [6]. However, DevOps makes one core assumption, namely that developers want to take part in operational activities. As discussed by Mike Gualtieri [34], this assumption might not always be true. Hence, this fact led to the apparition of the NoOps idea (No Operations).

NoOps is a term coined in 2011 and it refers to the promise that developers would spend their time only on development activities, without having to get involved in operational activities [34]. At this moment, NoOps is still a vague concept - there is no formal definition and no definitive set of characteristics. Furthermore, currently there is no academic literature that investigates NoOps, despite the fact that NoOps is an increasingly popular topic in online media.

Considering the fact that NoOps is a new trend in the software engineering practice, as well as the lack of academic literature concerning the topic, a survey of the NoOps-related ideas and practices is of high importance for both practitioners and academic research. Therefore, this thesis aims at providing an overview of NoOps through a systematic literature study, which is supported by rigorous qualitative and quantitative research methods. The study focuses primarily on the analysis of practitioners' literature, aiming at bridging the gap between industrial practice and academic research.

1.1 Research questions

Given the lack of academic literature regarding NoOps, the main source of knowledge is practitioners' literature, which is also commonly referred to as "gray literature" [30]. With this in mind, the scope of this research can be formulated as follows: "the analysis of gray literature sources with the purpose of identifying the different dimensions of the NoOps definition, its positioning, its associated characteristics, as well as the reported advantages and challenges related to the adoption of NoOps, from the perspective of the various contributors to the public discourse". This goal is aligned with that of similar works which investigate related concepts such as DevOps, DevSecOps or BizDevOps [18, 63, 58].

Hence, the following research questions are formulated:

RQ1. How is NoOps described in practitioners' literature?

RQ2. What are the characteristics of NoOps reported in practitioners' literature?

RQ3. What are the challenges of NoOps that are reported in practitioners' literature?

RQ4. How is NoOps described in online discussions?

The goal of RQ1 is to obtain an overview of the different dimensions of the NoOps definition and positioning, as well as the reported advantages associated with the adoption of NoOps. RQ2 aims at providing a taxonomy of technical and cultural characteristics associated with the concept, while the goal of RQ3 is to provide a survey of the main challenges, pitfalls and costs related to NoOps. Finally, RQ4 aims at analysing NoOps-related online discussions, in order to capture a snapshot of the practitioners' viewpoint. Since practitioners' literature can sometimes be published by companies for promotional purposes, we can hypothesize that this kind of literature might not be an accurate representation of reality. Hence, understanding the practitioners' viewpoint enables us to perform a comparison of the findings obtained from practitioners' literature with the perspective of practitioners.

1.2 Report structure

The rest of the report is structured as follows. Chapter 2 introduces related work. Then, chapter 3 introduces the systematic literature study, focusing on the depiction of NoOps in practitioners' literature. Then, chapter 4 presents the analysis of online practitioner discussions. Finally, chapter 5 outlines some conclusions and introduces some directions for future work.

Chapter 2

Related Work

Considering that the NoOps concept is still in its infancy, there is currently no academic literature that directly studies NoOps. Hence, this section will provide a positioning of the current work within the larger spectrum of software engineering research, and will introduce some studies with similar goals, but applied in different domains.

Overall, the current work is a systematic literature study with a primary focus on gray literature, using a mixed-methods approach. Due to the wider scope of this work and the research goal of providing an overview of a new concept in the state-of-the-practice, it can be classified as a systematic literature mapping study rather than a review, following the classification of Petersen et al. [71]. Due to the fact that NoOps was conceived as a response to DevOps, the study can be positioned in the research area of DevOps-related approaches, such as DevSecOps [63] or BizDevOps [58].

DevOps Systematic Literature Studies

Since NoOps was conceived as a response to DevOps, we introduce some studies that define this related area.

One of the main works that aims at characterizing DevOps is that of Jabbari et al. [42]. This study is a systematic literature mapping, focusing strictly on peer-reviewed literature. The work proposes a definition of DevOps, a taxonomy of related practices and a comparison of DevOps to other development methodologies. Therefore, the primary goals of this study are aligned with those of the current NoOps study, despite the differences in focal concepts. However, even though the primary goals are similar, the analysis methods employed in these studies are rather different. One difference is that the qualitative analysis conducted by Jabbari et al. relies solely on open coding for the conceptual analysis, while the qualitative analysis conducted in the current NoOps study uses an eclectic coding strategy.

Another major systematic literature study aims at providing a formal conceptualization of the DevOps culture [80]. Since software development methods, such as DevOps and Agile, are often considered to be cultures rather than sets of practices and tools, the authors wanted to provide a clear overview of what the DevOps culture entails. The study is based on 23 primary studies, all being considered academic literature. Their analysis method is based on open coding, memoing and constant comparison, which is fundamentally different than the analysis strategy employed in the current study. The results show that the DevOps culture can be characterized by a number of factors, the most important being communication, collaboration, feedback, responsibility, and knowledge sharing.

Other systematic literature studies are focused on DevOps augmentations. For example, Myrbakken et al. [63] investigate the concept of DevSecOps. They provide an overview of what DevSecOps is, the associated benefits and the challenges that organizations face during adoption. Another example is the paper of Lohrasbinasab et al. [58], which investigates an augmentation of

DevOps which is called BizDevOps through a multivocal literature review. Similar to the previously mentioned study on DevSecOps, this study aims at identifying the definition of BizDevOps, its characteristics, motivating issues, challenges and potential adoption benefits.

Systematic Literature Studies in Software Engineering

In software engineering research, systematic literature studies are a form of evidence aggregation, as used within evidence-based software engineering [47]. There are three main approaches for systematic literature studies: systematic literature reviews [48], systematic literature mapping studies [71], and snowballing [90]. One of the main tertiary studies investigating systematic literature studies in software engineering is that of Kitchenham et al. [47]. In this study, the authors explore the activity of systematic literature reviews in software engineering in terms of number of published studies, addressed research topics and limitations. One of the most important findings of this study is that most systematic literature reviews focus on the research practice, rather than specific software engineering practices. Hence, at the time of publishing, systematic literature reviews were more useful for researchers, rather than for practitioners.

In another paper, Garousi et al. [29] discuss the problem that systematic literature reviews in software engineering do not typically offer good representations of the state-of-the-practice, highlighting that there is a gap between academic research and industrial practice. As a solution, the authors propose the use of multivocal literature reviews, which are systematic literature reviews augmented with gray literature. Since gray literature is generally created by practitioners or for practitioners, multivocal literature reviews could be used to bridge the gap between research and practice. In the same study, the authors found that using gray literature in systematic literature reviews is not straightforward. Since gray literature is often based on the experience and the opinion of the author, the quality of this kind of literature is questionable, which could also affect the quality of the resulting multivocal literature review. Hence, in a follow-up study, Garousi et al. [30] propose a set of guidelines for conducting multivocal literature studies, including a set of recommendations for verifying the quality of gray literature sources. These guidelines have also been utilized while conducting the current NoOps literature study.

Kamei et al. [46] conducted a tertiary study with the goal of understanding how software engineering researchers use gray literature in their studies. One of the most important findings of the study is that most gray literature is created by companies and consultants. In addition to this, most of the studies analyzed in this research did not employ specific criteria for assessing the quality of the gray literature, which the authors find concerning, given the nature of this type of literature. One challenge that the authors identify related to the wider adoption of gray literature is the lack of reliability of the sources. The proposed solution is the use of gray literature originating from high or moderate outlet control of the content production. This includes types of sources such as books, magazines, whitepapers, or annual reports, among others.

One popular systematic literature mapping study that follows the guidelines of Garousi et al. [30], similar to our NoOps study, is that of Soldani et al. [83], which investigates the technical and operational benefits and difficulties encountered by practitioners with the microservice architecture. The authors selected 51 different gray literature sources, using the recommendations of Garousi et al. for literature search and quality-based sample selection. In addition to the main study findings, the authors also present their observations regarding the use of gray literature in systematic literature reviews. For instance, they claim that multivocal literature reviews are a valuable tool for exploring new areas of software engineering characterized by a high industrial adoption. Another remark is that the quality of gray literature is hard to measure, given the lack of a common format in gray literature sources. Hence, these quality control difficulties could affect the reproducibility of studies.

Systematic Literature Studies using automated analysis methods

Since the current study involves both manual qualitative and automated quantitative analysis techniques, this section will introduce two studies that follow similar approaches and which have been used as reference in the current work.

Cascavilla et al. [15] conducted a multivocal literature review, providing an overview of cyber threat intelligence. Similar to our study, this review also includes gray literature, following a methodology inspired from [83]. In this work, the data analysis is conducted using a mix of qualitative and quantitative methods, including thematic coding and topic modeling. One interesting insight from this paper is the fact that there is a clear overlap between the topic modeling and thematic coding results, from which it can be inferred that topic modeling could be a valuable tool for automated analysis in systematic literature studies.

Another systematic literature review with a focus on gray literature is that of Kumara et al. [50]. This literature review provides a taxonomy of good and bad practices for the DevOps approach of Infrastructure-as-Code. Although the study has been conducted primarily using qualitative techniques, a quantitative approach based on applying natural language processing methods was attempted as well. In particular, the authors used topic modeling and topological data analysis to perform an automatic analysis of the data. However, due to the lack of structure in the data, preprocessing difficulties arose, which resulted in low-coherence topics. Hence, the authors decided to focus only on manual qualitative analysis.

Chapter 3

NoOps in practitioners' literature

The goal of the first three research questions is understanding the various dimensions of the NoOps definition, the positioning, the associated characteristics, as well as the reported advantages and challenges related to the adoption of NoOps. To answer these questions, a gray literature study is conducted, considering the lack of academic literature related to NoOps. In other words, in this study the primary sources of knowledge originate from gray literature. Gray literature is often referred to as practitioners' literature and it can be defined as "any document that hasn't gone through peer review for a publication" [25]. Examples of practitioners' literature include sources such as technical reports, lectures, blog posts, news articles, videos or podcasts. This study was conducted following the guidelines of Petersen et al. [71] for conducting systematic literature mapping studies, as well as the guidelines of Garousi et al. [30] for including gray literature in systematic literature studies. The recent multivocal literature mapping study of Soldani et al. [83] has also been used as reference.

An overview of the process used to conduct this systematic literature study (after the formulation of the research questions and the preliminary planning of the study) is shown in Figure 3.1. The process begins with the search for primary sources using a variety of search strings on multiple search engines. The search results are then subjected to a selection procedure, which involves the application of a number of inclusion and exclusion criteria. Afterwards, forward snowballing is applied on the selected sources in an iterative manner. In other words, all sources that are referenced within the selected primary sources and that are not yet part of the set of selected primary sources, are added to a pool of candidate primary sources. This pool of candidate sources is then also subjected to the selection procedure. The snowballing and selection procedures are applied until no more primary sources could be found. Then, this final set of studies undergoes a quality assessment procedure, followed by data extraction and analysis. In the end, the analysis results are synthesized and the findings are reported.

It is important to note that parts of this study have been conducted during the seminar course of the Software Engineering & Technology research group. In particular, the definition of the search strategy, data collection, sample selection and quality control phases, as well as a small part of data analysis have been conducted during the seminar. The data analysis results that emerged from the seminar work have been used as a basis for the further analysis of the data (in the form of a pilot study). The seminar report can be found in Appendix H.

The rest of this chapter is structured as follows. Section 3.1 introduces the search strategy and the entire process for data collection. Section 3.2 provides a description of the sample selection and quality control phases, while section 3.3 introduces the data extraction and analysis processes. The findings are illustrated in section 3.4, while section 3.5 describes the inter-rater reliability process. Finally, section 3.6 provides a discussion of the results and section 3.7 introduces some potential threats to the validity of this study.

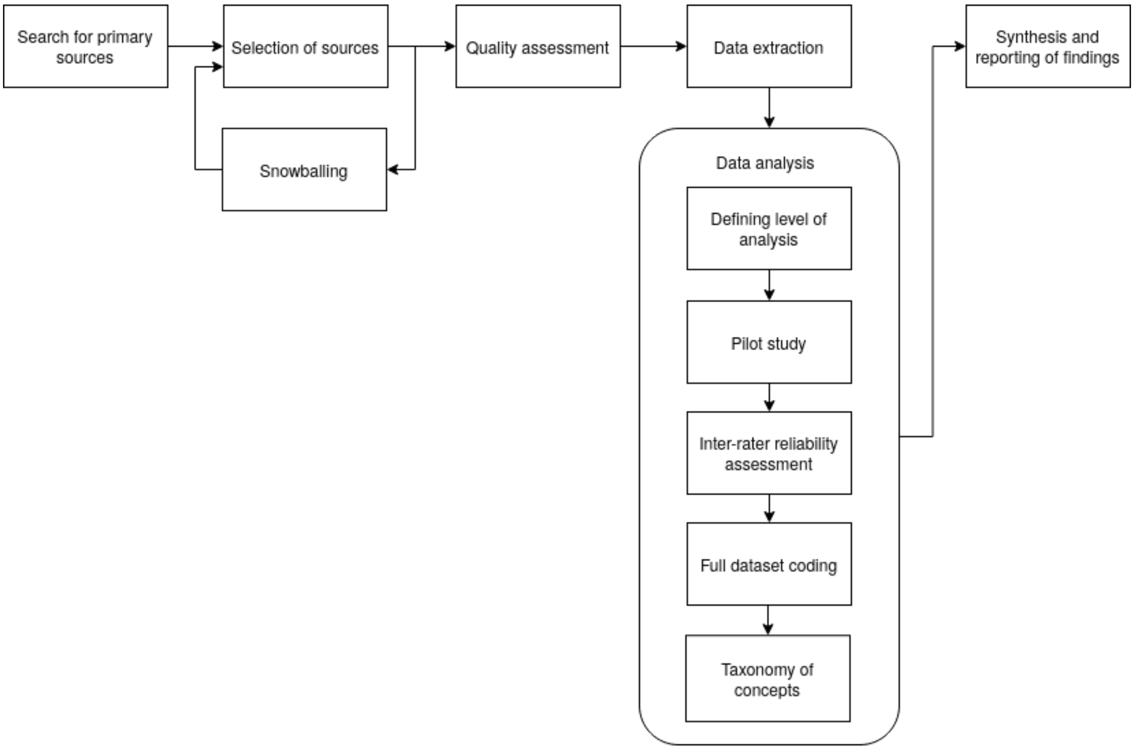


Figure 3.1: The general workflow for conducting the current systematic literature study

3.1 Search strategy and data collection

This section introduces the literature search strategy and data collection process. Please note that this section was entirely extracted from the seminar report (see Appendix H, section 3.2).

Based on the recommendations of Garousi et al. [30], gray literature can be found using mainstream search engines, such as Google, Bing or Yahoo. For the scope of this study, the decision was to use Google, Bing and Google Scholar as primary search engines. Google and Bing have been chosen due to their high popularity and usage in the English-speaking world. Google Scholar has been chosen in order to find gray literature related to NoOps that might have been referenced in peer-reviewed content, according to the guidelines of Yasin et al. [92]. Using the guidelines of Kitchenham et al. [48], a number of search strings have been created based on the main keywords of each research question, as listed below:

("NoOps" OR "No-Ops" OR "No Ops") AND ("definition" OR "characteristics" OR "practices" OR "culture" OR "challenges")

The search strings above have been used identically on all three search engines. In addition, the search parameter "Only English results" was used on all three search engines. The search has been conducted on December 8, 2020.

For each search string, the first 20 pages of results for each search engine have been considered. Since all the three search engines list the results in the order of relevance, the assumption was that all results returned after the first 20 pages would be very irrelevant.

After obtaining all search results, the duplicates were merged based on the URL.

3.2 Sample selection and quality control

This section introduces the sample selection and quality control process, which was conducted during the seminar. Parts of this section were extracted from the seminar report (see Appendix

H, section 3.3).

After performing the literature search, the next step was to manually filter the results based on multiple criteria. This filtering is necessary to identify those search results that are actually relevant to the goals of this study. The manual filtering was based on a set of inclusion and exclusion criteria, as shown in Table 3.1.

Inclusion	i_1. Source is in English i_2. Source provides an explicit discussion of NoOps as set of practices within software engineering i_3. Source is located on the first 20 pages of search results i_4. The content of the source is related to at least one of the research questions
Exclusion	e_1. Sources that are behind a paywall (excepting scientific publications) e_2. Sources that are completely product or service advertisements, without providing any additional insights related to NoOps e_3. Sources without a clear publication date e_4. Sources without explicit authorship e_5. Sources that are not available (do not exist) e_6. Sources that are duplicates

Table 3.1: Inclusion and exclusion criteria

The filtering process started with the application of the inclusion criteria. These criteria have been applied sequentially, one-by-one, with the exception of i_3, which was already applied at the search time. The inclusion criteria have been applied based on the information presented in the title and the meta-description of the search results in the case of $i_1 - i_3$, but also on the entire source content in the case of i_4.

After the application of the inclusion criteria, the filtering process continued with the application of the exclusion criteria. These criteria have also been applied sequentially, one-by-one, starting from e_1. Please note that criterion e_3 has been considered necessary because it would enable the mapping of results and sources to a specific timeline. Moreover, e_4 has been considered because the author names are required in the quality control stage, in order to establish their level of expertise.

Finally, the resulted list of primary studies has been subjected to quality control, following the recommendations of Garousi et al. [30]. The reason why additional quality control is recommended for gray literature studies, in contrast to traditional systematic literature studies, is that gray literature is not peer-reviewed, therefore the conveyed evidence can not be fully trusted.

In opposition to the approach of Garousi et al., our quality control does not aim at eliminating sources based on a quality threshold. In fact, the goal of our quality control is to sort the primary sources in descending order, based on the quality score. This sorted list is used to define the order of analysis for the sources. The assumption, in this case, is that this form of purposive sampling would aid the process of reaching theoretical saturation early in analysis phase, since it is expected that the primary sources that have a higher quality score would also provide more in-depth results.

The quality control has been applied based on a variant of the quality checklist proposed by Garousi et al. [30]. To be more specific, the authors proposed a quality checklist with 20 different items, aiming at assessing various quality criteria. These 20 proposed items aim at evaluating the quality of a source through eight different criteria: authority of the producer, methodology, objectivity, date, position with respect to related sources, novelty, impact and outlet type. The authors highlight that the main limitation of this checklist is that not all items are suitable for all kinds of gray literature sources, i.e., the checklist can not be used out-of-the-box. Hence, the authors recommend tailoring the quality control checklist and selecting only the items relevant for each study.

From this checklist, we have selected 6 items that could be applied for this study and adapted them to the current context. In particular, in the quality control phase the goal is to answer the following questions:

Q_1. Is the author of the source a practitioner?

Q_2. Does the publisher review content before publication?

Q_3. Is the author associated with a company or organization developing software or providing software-based services?

Q_4. Does the author generally publish works in the field of DevOps, Serverless, NoOps, ITOps, or Software Engineering in general?

Q_5. Does the source provide any form of evidence for the claims?

Q_6. Is the source type reputable?

In Q_1, by practitioner we refer to any job related to software engineering, DevOps, or cloud computing, even if it is a managerial or a technical job. The goal of this question is to determine whether the public discourse reflects a real practitioner perspective, rather than the perspective of non-technical individuals, such as journalists or content writers. In Q_2, the goal is finding out whether the publishing outlet performs any sort of content reviewing before publication. For instance, while magazine articles and whitepapers are likely to be reviewed before publication, this might not be the case for ad-hoc blog posts. Q_3 and Q_4 aim at investigating whether the author has any relation to a specific software company and if the author generally publishes other works related to software engineering. Then, Q_5 aims at investigating whether the presented claims are supported by any form of evidence. In particular, the focus of this question is on the presence of references or any type of evidence described in the hierarchy of Alves et al. [3]. Finally, Q_6 is assessed based on the reputability guideline of Garousi et al. [30], which implies that the sources which have a high outlet control and credibility (e.g. books and whitepapers) have a higher quality than sources that have a moderate or low outlet control and credibility (e.g. blog articles or podcasts). Please note that the items Q_1 to Q_4 are answered by checking the online presence of the author via search engines and social media platforms.

As discussed in the paper of Garousi et al. [30], the next step after addressing the items above is to compute an overall quality score as follows. For all the items included in the checklist, a score of either 0 or 1 is assigned (1 if the answer to a question is yes, 0 otherwise). However, in the case of the source type reputability item (Q_6 in the current context), the score is assigned following the scoring scheme of Garousi et al. [30]: 0 if the source has a low outlet control (e.g., blogs), 0.5 for sources with moderate outlet control (e.g., news articles, annual reports) and 1 for sources with high outlet control (e.g., books, magazines). Then, an average for all these scores is computed per primary source, obtaining a normalized quality score. After obtaining the quality score for each source, the entire list of primary sources is sorted in a descending order based on this normalized quality score (in contrast to the approach of Garousi et al. [30], which enforces source elimination).

3.3 Data extraction and analysis

In the data extraction phase, the list of primary sources was first sorted by quality score in a descending order. Then, for each textual primary source, the content was extracted per paragraph into an Excel worksheet. In the case of video and audio sources, the content has been transcribed based on the "chunk transcription" method presented in [10]. Moreover, annotation coding is performed [79] in order to record whose voice is reflected (authors' occupation), as well as the type of organization reflected in each primary source. Annotation coding was used to extract also the publication date, source type (e.g., scientific paper, news article, conference talk, etc) and type of publishing outlet.

In order to answer the research questions, content analysis is performed on the selected sources. According to Krippendorff [49], "content analysis is a research technique for making replicable and valid inferences from texts (or other meaningful matter) to the contexts of their use". The two main types of content analysis are conceptual analysis and relational analysis. While conceptual analysis involves primarily the quantification of the presence of (implicit or explicit) concepts in a text, relational analysis focuses both on the quantification of the presence and the relationships among concepts in a text [66]. Since the goal of this study is to provide an overview of NoOps, we are mostly interested in what are its associated concepts and themes, rather than the relationships between these associated concepts. Hence, the focus of this study will be restricted to conceptual analysis.

The conceptual analysis steps involved in this study are presented below:

1. **Deciding the level of analysis**. This first step involves determining the appropriate unit of analysis. According to Krippendorff [49], the size of the analysis unit affects the efficiency of the descriptive effort. In other words, if the unit of analysis is too long (e.g., an entire article), the analysis process would yield a very general and unreliable result, since significant information might be lost in the inference process. However, if the unit of analysis is too short (e.g., a sentence), the analysis output might be affected by validity issues, since the analyzed unit might not provide a suitable context for a proper inference. For situations that require the characterization of a concept within a certain context, Klaus Krippendorff recommends choosing an analysis unit which is not larger than a paragraph, since this size would yield both reliable and valid results [49]. Following this recommendation, we decided to perform the analysis at the level of individual paragraphs.

2. **Pilot study**. After deciding upon the unit of analysis, a pilot study has been conducted. The purpose of this pilot study is to generate an initial set of codes (so called code book), that is later applied during the full dataset coding phase. The pilot study was conducted by one researcher and it involved the analysis of a sample of sources using Structural and Descriptive coding. The first 30 sources with the highest quality scores have been selected for this pilot study, under the assumption that by analyzing these highest-quality sources first, theoretical saturation would be reached quicker, so the resulted list of codes would provide an exhaustive reflection of NoOps.

   According to MacQueen et al. [60], structural coding means the application of a "content-based or conceptual phrase representing a topic of inquiry to a segment of data that relates to a specific research question". In the current study, structural coding was applied as follows. When an analyzed paragraph contained information related to one of our research questions, a code following the format "RQ#. variable of interest" was applied to that paragraph, where # represents the number of the related research question. For instance, if a paragraph contained a technical characteristic of NoOps, then the code "RQ2. Technical characteristics" was assigned. However, when a paragraph did not contain any information related to any of the research questions, the code "N/A" was assigned. The reason for using structural coding as a first step in the analysis is to provide an exploratory categorization of the information that could provide findings related to specific research questions.

   Since the findings from the structural coding are rather general, a more detailed analysis is conducted on these findings using Descriptive coding. According to Saldana [79], descriptive coding represents the application of a short phrase that represents the main topic of an analyzed text. In the current study, descriptive coding was applied as follows. For all paragraphs that have been classified as relevant during structural coding (i.e., not "N/A"), a very specific code that encompassed the topic of the paragraph was assigned. For example, if a paragraph was assigned the structural code "RQ2. Technical characteristics" and it discussed the use of Chaos Engineering methods in NoOps, the code "Chaos Engineering" was assigned to the paragraph. It is important to note that these descriptive codes were created spontaneously during the analysis process, i.e., no predefined categories were used.

3. **Inter-rater reliability assessment**. Considering that the work in this study is conducted by a single person, and since qualitative research can be considered of subjective nature, significant biases might be introduced during the coding processes. Hence, in order to detect

and mitigate these biases early on, inter-rater reliability (IRR) measurements are computed on a sample coded by two different people based on the code book that emerged from the pilot study. Both coders are Computer Science-trained master students, specialized in Software Engineering. The IRR results were used to detect disagreements between the two coders and mitigate the root cause of these disagreements. The inter-rater reliability assessment process and results are further presented in section 3.5.

4. **Full dataset coding**. After the IRR assessment and resolution of disagreements, the remaining sources have been coded by one researcher, using the list of structural and descriptive codes obtained from the pilot study (the code book). However, since the code book was created based on a pilot study, there was no guarantee that the sample analyzed in this pilot study was exhaustive and representative for the entire dataset. Hence, to account for this possibility the following system was applied: when a new code would be discovered, it would be marked as "Other". After the entire dataset is analyzed, all the codes marked as "Other" would be assigned specific meanings. Then, all the codes previously marked as "N/A" would be re-analyzed, hence accounting for the newly created codes.

5. **Taxonomy of concepts**. Once the full dataset coding was completed, the taxonomy of concepts was created using an approach called card sorting [93]. Card sorting is a process for assigning higher-level categories to fragments of text. This method involves placing the fragments of text (descriptive codes) onto individual pieces of paper (cards), which are iteratively sorted into groups. In the current study, open card sorting was used, therefore the groups emerged naturally during the sorting process. In other words, open card sorting implies that no predefined groups are used during the sorting process. While performing the card sorting, the guidelines of Zimmerman have been used [93]. Finally, the resulting taxonomies were represented in the form of mind map diagrams.

3.4 Findings

This section will introduce the findings of the current systematic literature study.

3.4.1 The search for primary studies

The findings presented in this subsection are identical to those from the seminar report (see Appendix H, section 4.1).

Appendix A illustrates the number of results obtained for each search string per search engine. All the results have been collected in a single day, on December 8, 2020.

Overall, 2182 results have been returned. After merging the search results based on their URL, 971 unique sources remained. These sources have then been subject to the application of inclusion and exclusion criteria. The statistics resulted from the application of inclusion criteria can be found in Table 3.2. As it can be observed, after the application of all inclusion criteria, only 133 sources remained. Please note that the reason why the application of criterion i_3 did not result in any difference is that this criterion was applied at the search time. After the application of exclusion criteria on these remaining sources, 95 sources remained. On these 95 primary sources, forward snowballing has been applied in two iterations, revealing 3 additional sources. The final list of 98 primary sources can be found in the bibliography.

Inclusion criterion	Description	# before	# after
i_1	Source is in English	971	844
i_2	Source provides an explicit discussion of NoOps as set of practices within software engineering	844	286
i_3	Source is located on the first 20 pages of results	286	286
i_4	Source is relevant to at least one of the research questions	286	133

Table 3.2: The number of primary sources before and after the application of each inclusion criterion

3.4.2 Overview of the selected primary studies

In this subsection, an overview of the selected primary studies is provided. The results in this subsection are based on the preliminary data annotations, which can be found in Appendix B.

Figure 3.2 illustrates the distribution of the sources based on the publishing year. The first primary sources related to NoOps were published in 2011. 10 sources were published in 2012, while in 2013 there were no published sources. The number of published sources started increasing again in 2014 reaching a peak in 2019, when 26 sources have been published. In 2020, 19 sources were published. However, since the search was conducted in late 2020, the data for 2020 might be incomplete. Moreover, the figure shows a positive trend, signaling that NoOps is becoming more popular in practitioners' literature.

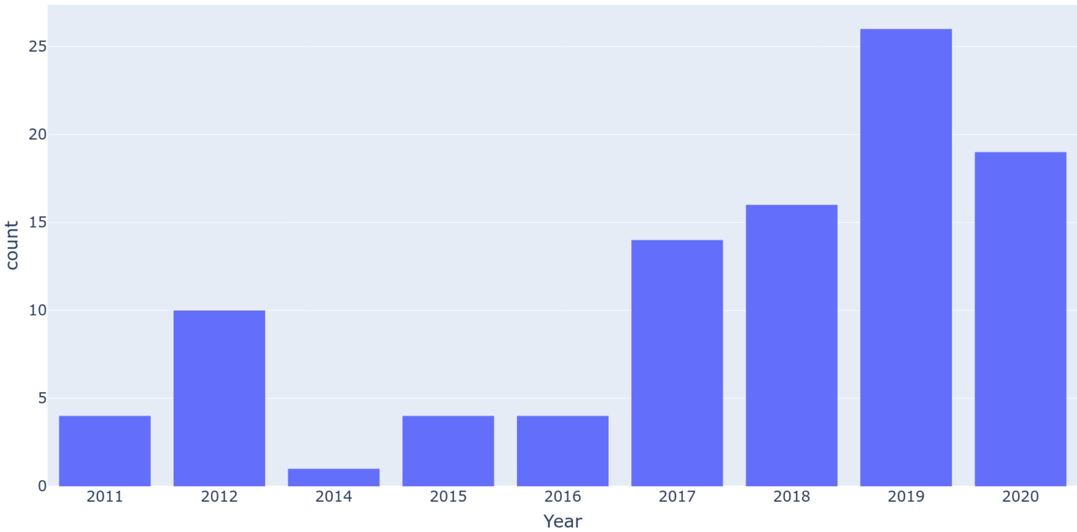


Figure 3.2: The number of sources published per year

Figure 3.3 presents an overview of the types of primary source publishers. Eight types of publishers have been identified: companies, book publishers, magazines, ad-hoc blogs, communities, independent podcasts, practitioner conferences and scientific journals. While companies seem to have published almost half of all considered sources (e.g. on company websites), approximately a quarter of all sources have been published in communities (platforms that allow anyone to contribute). The fact that almost half of the sources were published by companies shows that NoOps is a significant concept in the industrial practice. This also confirms the findings of Kamei et al. [46], that most gray literature is published by companies. However, this could also imply that the objectivity of the data in these sources is questionable, which could affect the quality of the results. This issue will be further discussed in the upcoming sections. Another observation is that the high number of sources published in magazines and communities might imply a considerable interest of practitioners in NoOps. Finally, the publication trends per type of publisher are presented in Figure 3.4. It is important to note that starting from 2016, the number of company contributions increases sharply, fact that strengthens the previous observation that NoOps could be a significant concept in the industrial practice.

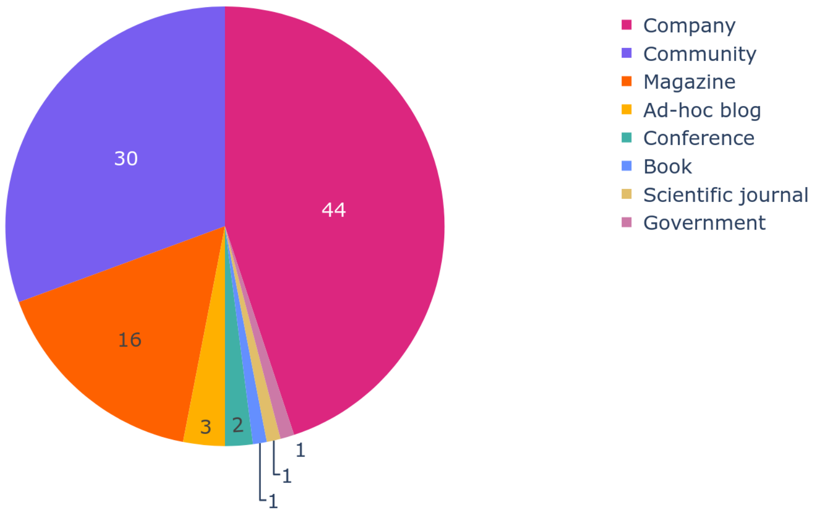


Figure 3.3: Overview of publisher types

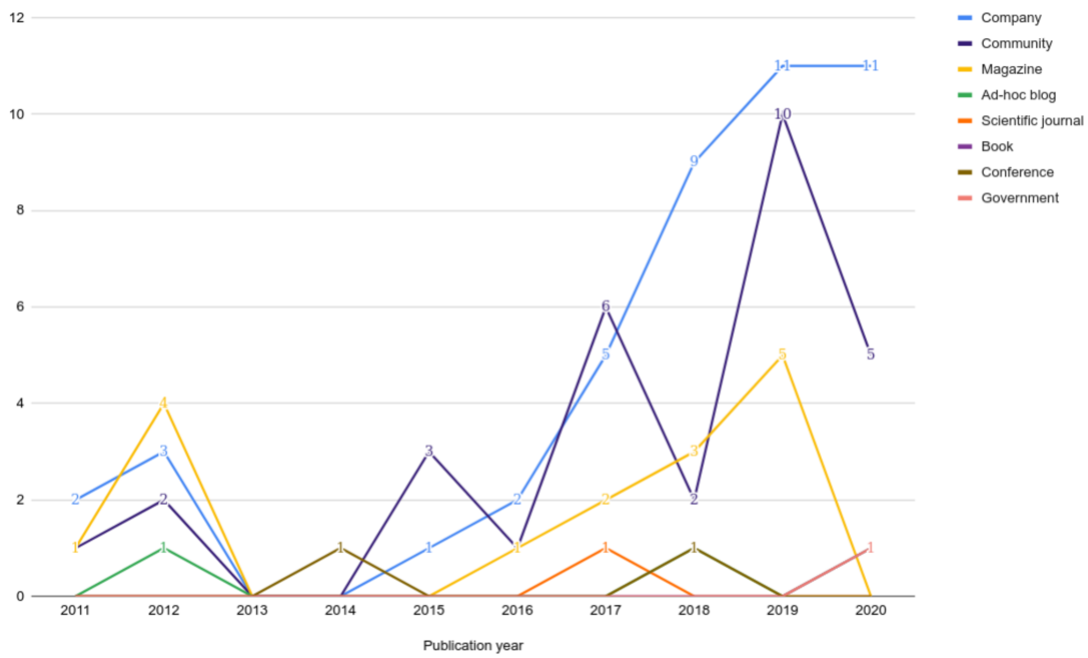


Figure 3.4: The number of primary sources published per year (by type of publisher)

Regarding the types of sources, as shown in Figure 3.5, the following types have been identified: blog articles, podcasts, whitepapers, books, magazine articles, papers and videos. While blog articles seem to be the majority, there is also a considerable number of whitepapers and magazine articles. The fact that blog posts represent the majority implies that the quality of the data is debatable. As discussed by Garousi et al. [30], blog articles are considered to be having a low outlet control and credibility, implying that the use of such sources could threaten the validity of the findings. The publication trends per type of source are presented in Figure 3.6.

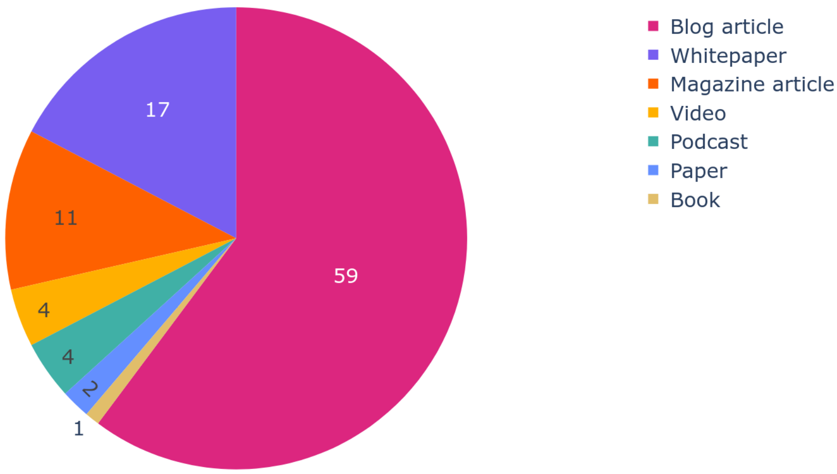


Figure 3.5: Overview of source types

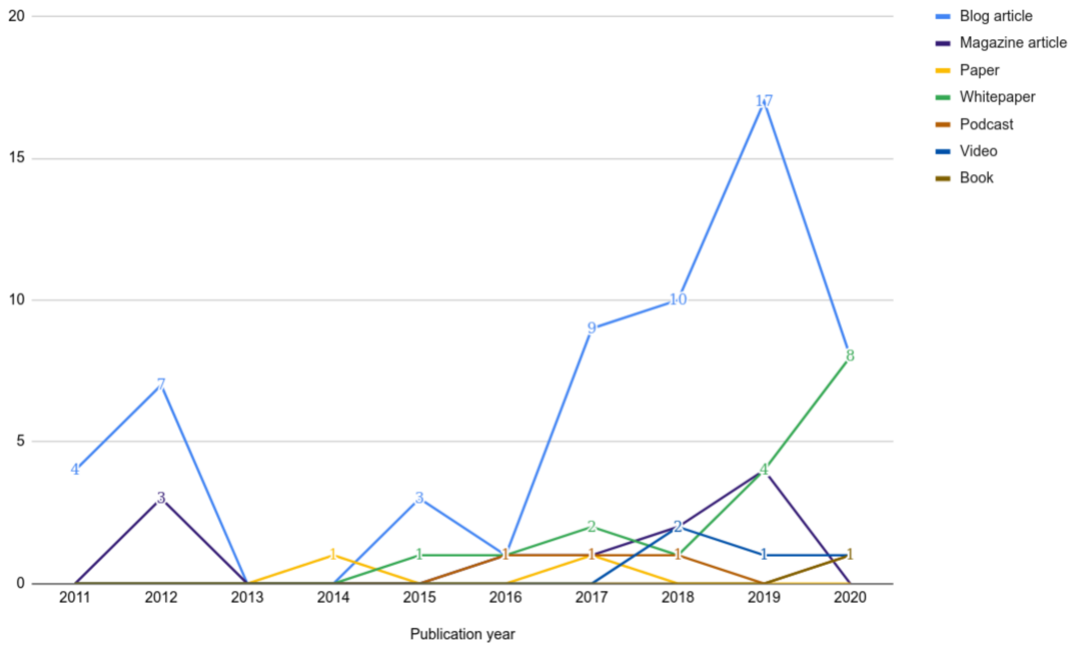


Figure 3.6: The number of primary sources published per year (by type)

Figure 3.7 illustrates the job types of all the authors of the selected primary sources. Six main categories of jobs have been identified: content creation, executive (all executive-level positions in an organization, e.g., CTO, CEO), research, marketing, engineering and collective authorship (consisting of all sources written on behalf of a group, without an explicit naming of the involved persons or an explicit voice reflection). The sources for which the author's activity domain could not be identified were marked as "-". It is interesting to note that most of the sources (33%) were authored by people having an executive position in an organization, most of these being technical executives. Authors that primarily focus on content creation (e.g., writers and editors) wrote approximately 24% of the sources, while engineering-related authors composed about 17% of the sources. The classification presented in this figure will be used later in this study to create mappings between authors' expertise and certain findings, with the goal of investigating whether certain job categories are more likely to report specific items.

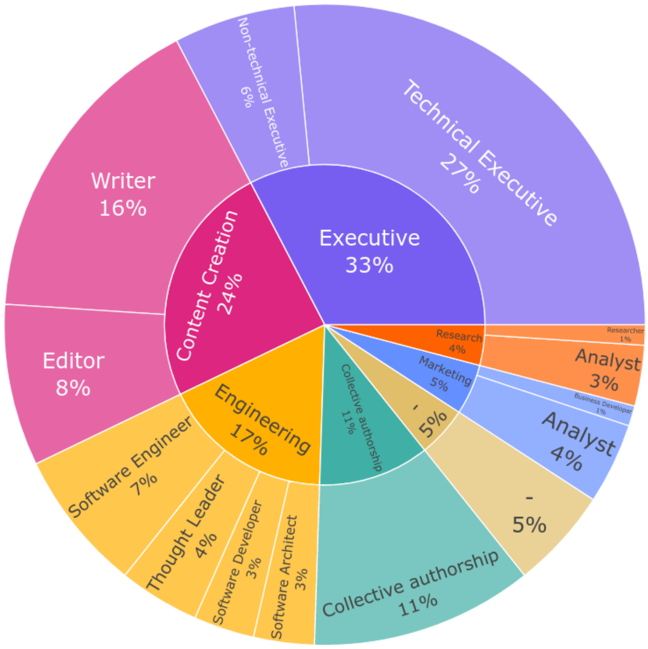


Figure 3.7: An overview of authors' job types for the entire pool of sources (the types of reflected expertise)

Figure 3.8 provides an overview of the types of organizations represented in the pool of selected sources. The following types of organizations were identified: consultancy, product development, publishing media, research, governmental organizations, marketing consultancies, and recruitment. Consultancy companies represent approximately 32% of the sources, while companies that focus on (software/cloud) product development represent 30% of the sources. It is interesting to note that about 23% of all sources are not representing the voice of specific tech-related companies, being written on behalf of publishing organizations (e.g., tech magazines). The remaining 15% is shared by research and governmental organizations, marketing consultancies and recruitment companies. The classification presented in this figure will be used later in this study to create mappings between company types and findings, in order to investigate whether certain organization types are more likely to emphasize certain findings.

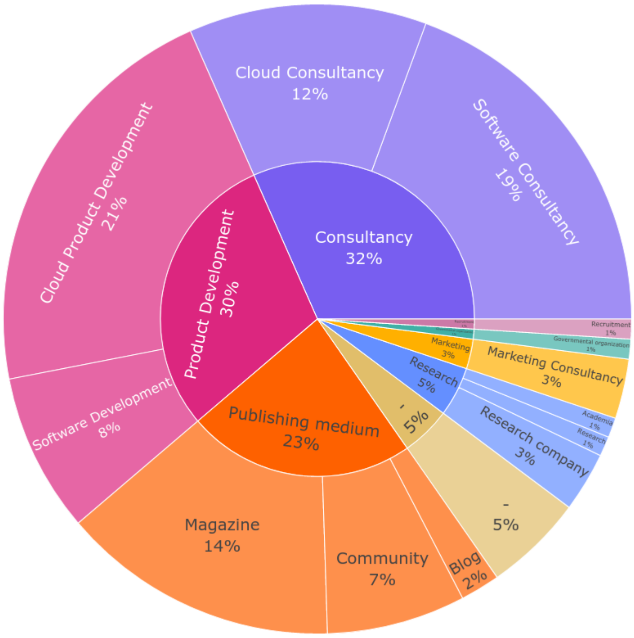


Figure 3.8: An overview of the types of organizations represented in the entire pool of selected sources

3.4.3 Findings emerging from the content analysis

In this subsection, the findings emerging from the analysis of primary sources will be introduced. A more detailed discussion of these findings is provided in section 3.6. The complete mapping of findings to primary sources can be found in Appendix C. Moreover, some figures highlighting the mapping between findings, and authors' job types and reflected organization types, can be found in Appendix E. Some of these figures will also be discussed in this subsection. All results are made available via a replication package.[1]

RQ1. How is NoOps described in practitioners' literature?

For this first research question, there are three main types of findings: NoOps definition, positioning, and advantages.

NoOps definition For the NoOps definition, 8 main perspectives have been identified in 75 different paragraphs, as it can be observed in Figure 3.9. The most prevalent definitions are that NoOps means developers do not have to interact with the operations team at all, or that NoOps is a mindset which implies that no traditional operational tasks are necessary to deploy and operate software. Other definitions define NoOps as the goal of completely automating the deployment, monitoring, management of applications and infrastructure, or that NoOps means the total exclusion of the operations teams. The other four perspectives are similar, highlighting that NoOps implies a highly automated IT environment, which could also be abstracted from the underlying infrastructure, such that a dedicated operations team becomes unnecessary. A tabular view of these different perspectives can be seen in Table D.1.

[1]Link to replication package: `https://github.com/drgs/noops-slm-reproc`

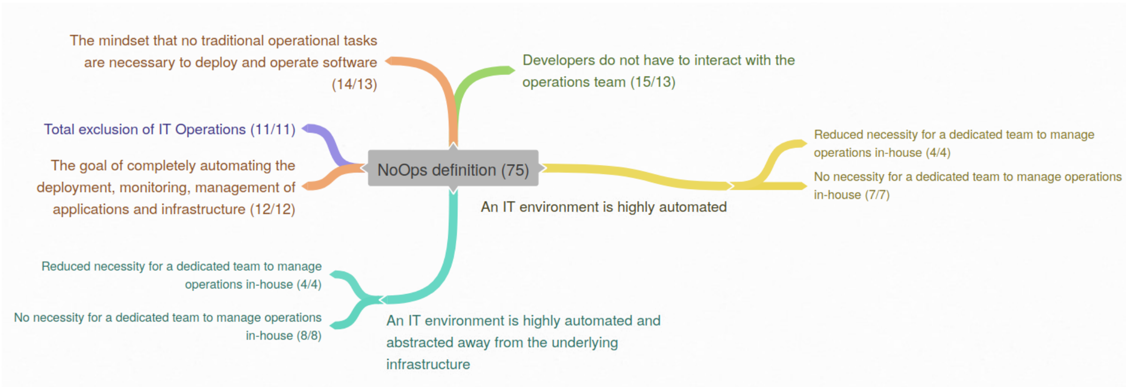


Figure 3.9: Mind map diagram highlighting the main dimensions of the NoOps definition. The first number in the brackets represents the number of paragraphs in the analyzed literature that reflect a certain dimension, while the second number represents the number of sources reflecting the dimension.

Figure 3.10 illustrates the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps definitions. It can be observed that product development and consultancy companies report all the perspectives, while governmental organizations report only one perspective. It is interesting to note that marketing-related companies do not report any definition of NoOps, while recruitment companies emphasize the perspective that "developers do not have to interact with the operations team". As discussed in a previous study [85], developers sometimes do not enjoy the cooperation with the operations teams, especially in the early stages of the development lifecycle. Hence, the fact that a recruitment company promotes this perspective could be a reason for attracting potential candidates, promising them an "operations-free" environment.

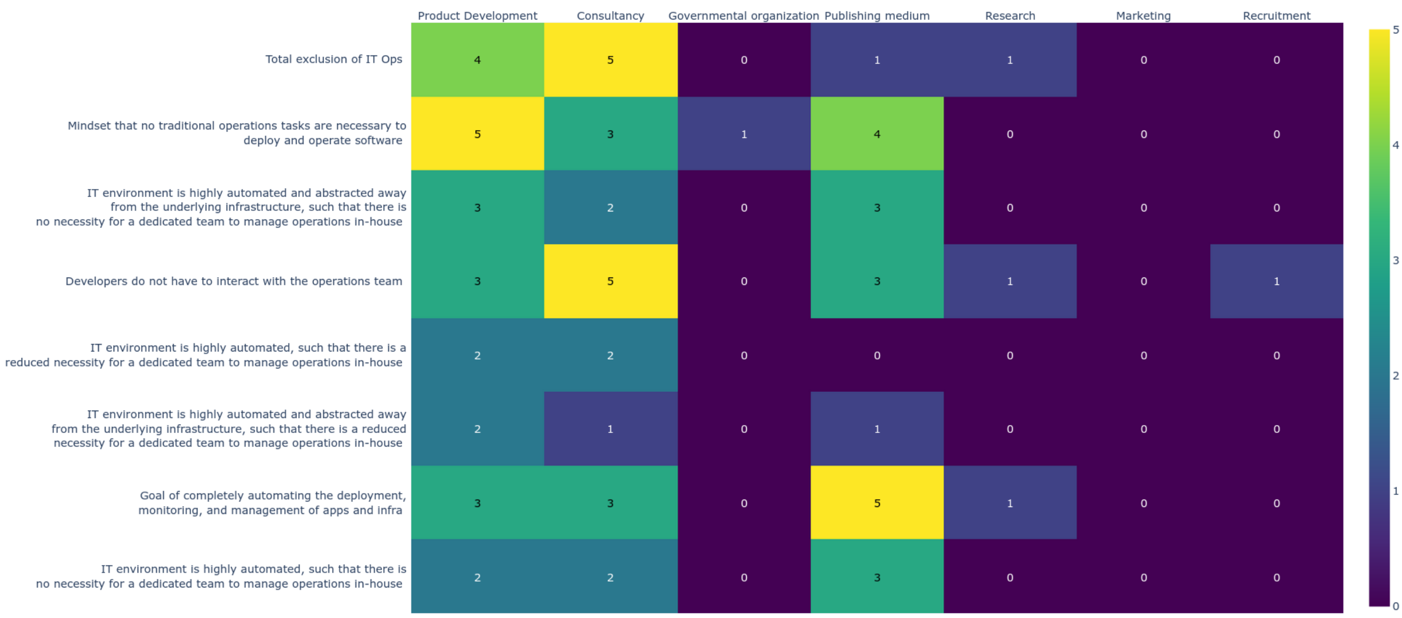


Figure 3.10: Heatmap showing the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps definitions. The number in each cell represents the number of organizations within one type reporting a certain finding.

NoOps positioning When it comes to the positioning of NoOps in relation to other practices, only one perspective has been observed in the analyzed gray literature, namely that NoOps is an augmentation of DevOps (see Table 3.3). This perspective has been mentioned 19 times, in 15 different sources.

Code	# paragraphs	# sources
NoOps is an augmentation of DevOps	19	15

Table 3.3: The frequency of codes related to the NoOps positioning

NoOps advantages The NoOps advantages reported in the analyzed literature are illustrated in Figure 3.11. This figure depicts the categorization of these reported benefits, which was achieved using the card sorting method. As it can be observed, the main categories of benefits are related to operational processes, organization and business, but also to social and technical aspects. The most commonly reported benefits are those related to social aspects, followed by operational processes, organizational and business, and those related to technical aspects. These benefits emerged from 96 different paragraphs. A tabular view of these findings can be seen in D.2.

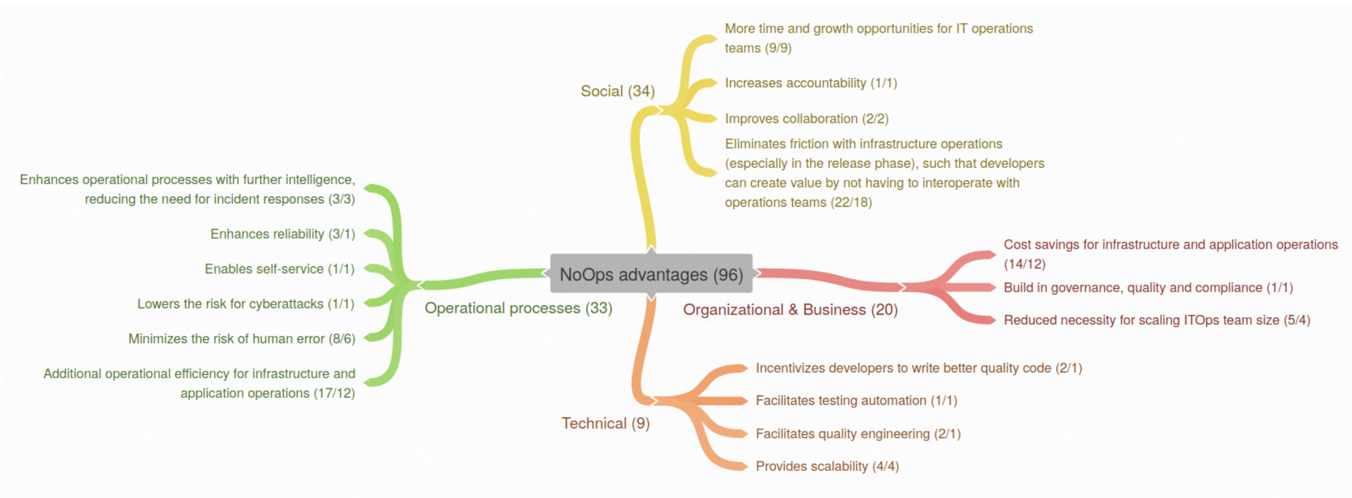


Figure 3.11: Mind map diagram highlighting the reported NoOps benefits. The first number in the brackets represents the number of paragraphs in the analyzed literature that reflect a certain benefit, while the second number represents the number of sources reflecting that benefit.

Figure 3.12 highlights the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps advantages. One important observation is that most advantages are reported by two types of companies, namely product development companies and consultancy. Moreover, marketing organizations do not report any benefits, which is surprising. As indicated by Pfleeger and Menezes [73], technology adoption decisions are made based on a variety of factors, one of the most important being technology's attributes. One of these attributes is represented by relative advantages, which means that potential adopters often need to understand the expected advantages before the actual adoption. Based on this fact, we can speculate that the NoOps-related sources published by marketing organizations were not aiming at promoting the adoption of NoOps, but rather aiming at attracting companies that already offer NoOps-related services, with the goal of providing marketing services. This could also explain why marketing companies did not report any NoOps definition (as seen in Figure 3.10). However, strong conclusions can not be derived, given the small number of sources published by this type of organizations. The suggestions of Pfleeger and Menezes [73], however, could explain why most benefits are reported by product development and consultancy companies. If these companies offer NoOps-related products or services, it could be the case that reporting NoOps advantages is a marketing strategy, aiming at raising the awareness about the possible benefits of NoOps, which could lead to attracting potential clients. Hence, one could wonder about what kind of evidence do these companies provide for their claims. Although this issue is not directly investigated in the current study, it could be an idea for future work. To account for the possible lack of evidence for the reported advantages, practitioners' viewpoint needs to be considered as well (this will be addressed in chapter 4).

Figure 3.12: Heatmap showing the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps advantages. The number in each cell represents the number of organizations within one type reporting a certain finding.

RQ2. What are the characteristics of NoOps reported in practitioners' literature?

The findings for this second research question can be organized in two main categories, namely technical and cultural characteristics. The findings related to the former category can be found in Table D.3, while those related to the latter can be found in Table D.4.

Technical characteristics The identified technical characteristics originate from 244 different paragraphs, and they were grouped in four main categories, namely automation aspects, tooling, practices, and architectural characteristics, as depicted in Figure 3.13. The practices category has been later split into two additional types, namely NoOps-specific practices, as well as General practices. The most commonly reported NoOps technical characteristics are related to practices, followed by automation aspects, tooling and architectural characteristics. A tabular view of these technical characteristics can be found in Table D.3

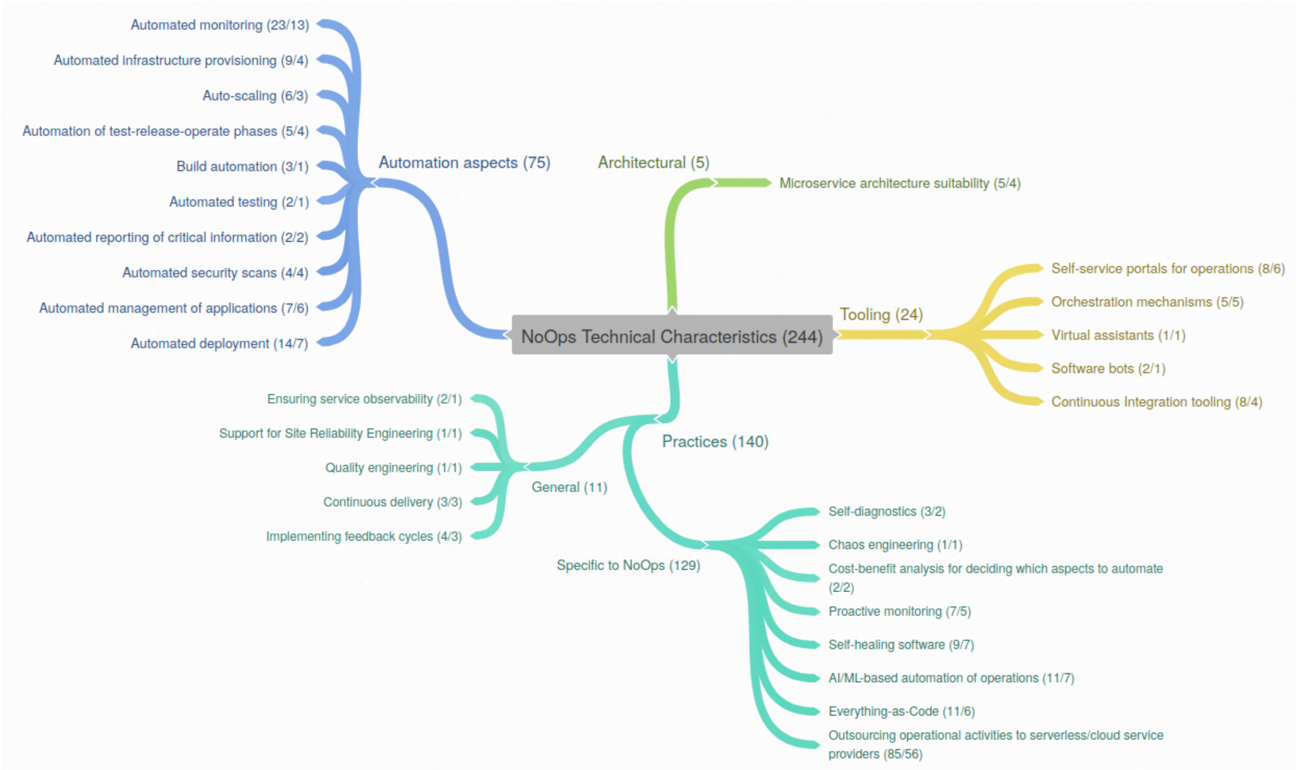


Figure 3.13: Mind map diagram highlighting the reported NoOps technical characteristics. The first number in the brackets represents the number of paragraphs in the analyzed literature that reflect a certain characteristic, while the second number represents the number of sources reflecting that characteristic.

Cultural characteristics In the current context, by cultural characteristic we refer to any characteristic that is related to the organizational culture. According to Denison [19], organizational culture is represented by organization-wide ways of working, habits and traditions. The NoOps cultural characteristics emerged from 48 different paragraphs, and they were categorized into four main types: collaborative, organizational, mindset and responsibility shifting, as shown in Figure 3.14. The most common type of cultural characteristics are the collaborative ones, followed by those related to responsibility shifting, organizational and mindset-related. A tabular representation of these findings can be found in Table D.4.

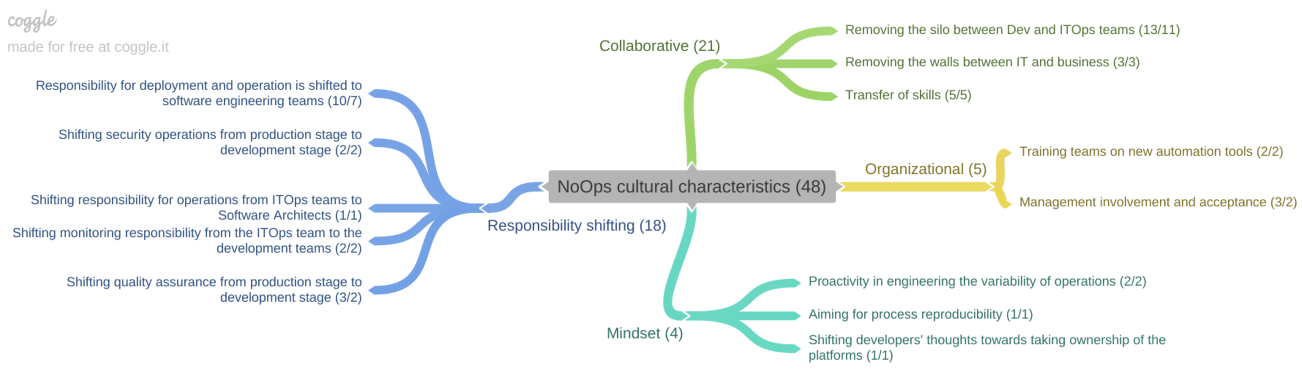


Figure 3.14: Mind map diagram highlighting the reported NoOps cultural characteristics. The first number in the brackets represents the number of paragraphs in the analyzed literature that reflect a certain characteristic, while the second number represents the number of sources reflecting that characteristic.

Figure 3.15 illustrates the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps cultural characteristics. It can be noticed that product development and consultancy companies report most of the characteristics, while research and marketing organizations do not report any. It can also be observed that recruitment companies support the characteristic of "proactivity in engineering the variability of operations". In a NoOps culture, operations teams are forced to redefine their role, moving from a "plumber perspective" (fixing issues after they happen) towards an "engineering perspective" (automating their tasks) [R13]. The thought of complete automation of operations often makes operations staff have the fear of losing their job [R88]. Hence, it could be the case that recruitment companies reported this characteristic as a signal for operations staff that they would still be relevant on the job market, since they are essential in helping organizations migrate towards a NoOps environment.

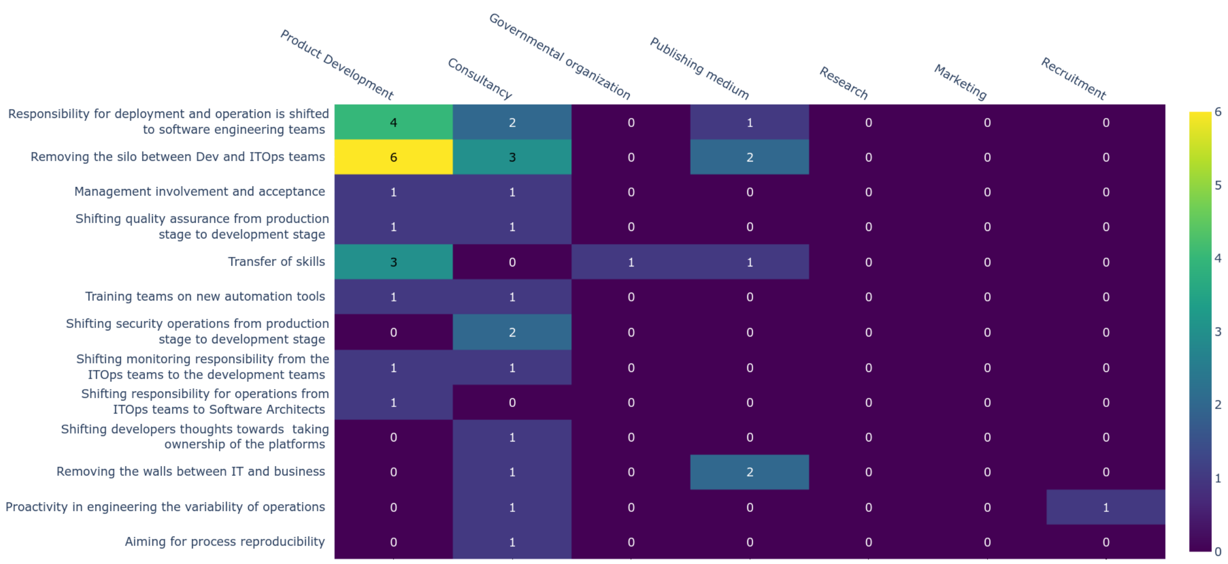


Figure 3.15: Heatmap showing the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps cultural characteristics. The number in each cell represents the number of organizations within one type reporting a certain finding.

RQ3. What are the challenges of NoOps that are reported in practitioners' literature?

Five types of challenges related to NoOps have been identified in the analyzed literature, the information being extracted from 69 different paragraphs. These types are: social, technical, cultural, organizational and general. Social challenges refer to any challenge of social nature (e.g., interpersonal communication, interactions, behavior) that is faced during or after the adoption of NoOps. Technical challenges are related to any challenge or problem of technical nature, while cultural challenges refer to any challenge or problem related to the organizational culture (following the previously-introduced definition of Denison [19]). Organizational challenges are related to any kind of challenge that affects the organization as an entity. Finally, general challenges are those challenges that could not be assigned to any other category.

A taxonomy of these challenges is depicted in Figure 3.16. A tabular representation of these challenges can be found in Tables D.5, D.6, D.7, D.8, and D.9.

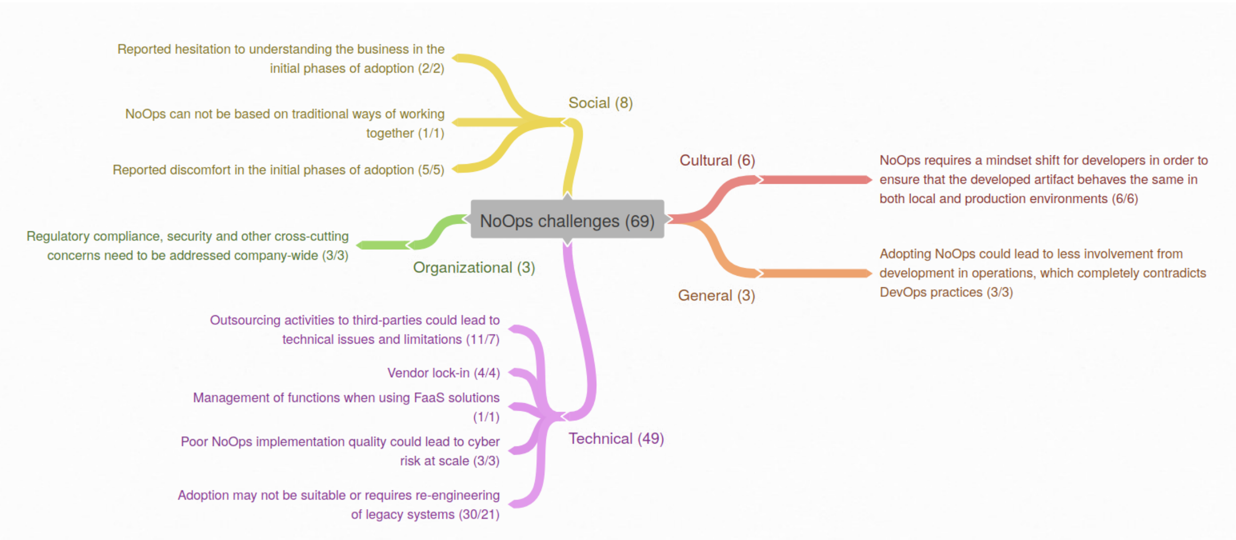


Figure 3.16: Mind map diagram highlighting the identified NoOps challenges. The first number in the brackets represents the number of paragraphs in the analyzed literature that reflect a certain challenge, while the second number represents the number of sources reflecting that challenge.

Figure 3.17 shows the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps challenges. Consultancy companies report most of the challenges, the most frequent being that the NoOps "adoption may not be suitable or requires re-engineering of legacy systems". Despite the fact that this challenge is reported by other types of organizations as well, the fact that consultancy companies report this challenge most frequently could represent a proof for the previously discussed concern, namely that the primary sources published by these companies might be part of a marketing plan, aiming at attracting potential clients. Moreover, it is interesting to note that all social challenges have been reported by product development companies, which could imply that these challenges might be based on their own experience, rather than on expectations.

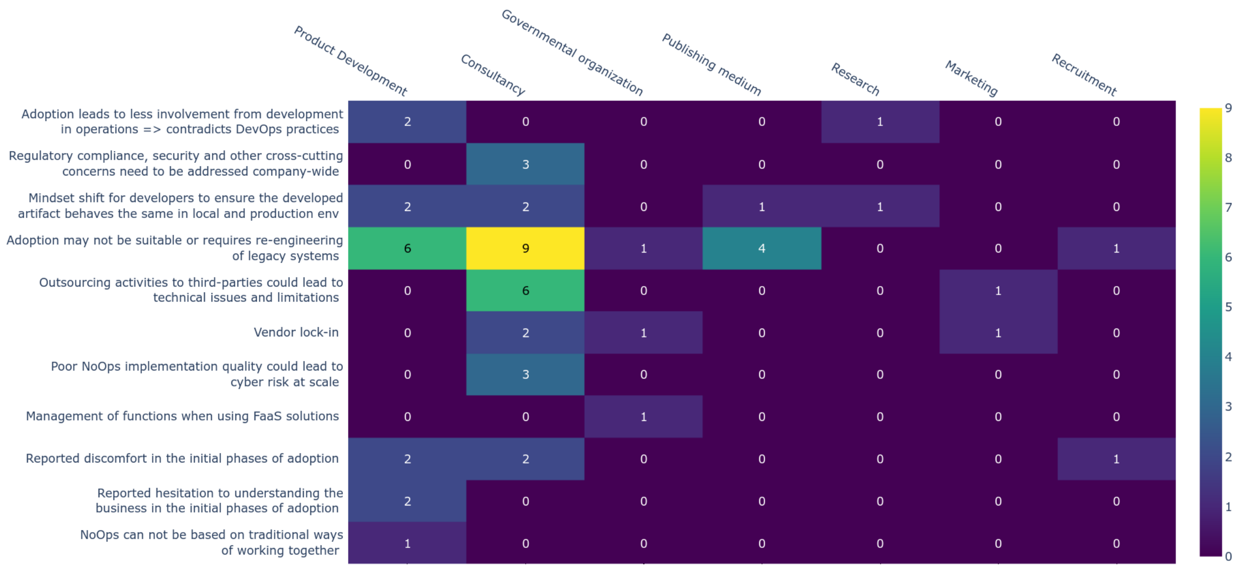


Figure 3.17: Heatmap showing the mapping between the types of organizations represented in the entire pool of sources and the reported NoOps challenges. The number in each cell represents the number of organizations within one type reporting a certain finding.

3.5 Inter-rater reliability assessment

As previously discussed in section 3.3, inter-rater reliability (IRR) assessment is performed in order to understand whether the coding performed by the main author is based on objective decision making factors. This section will present the results of this inter-rater reliability assessment and will provide a summary of the subsequent discussion for solving the coding disparities.

The process for conducting this IRR assessment can be described as follows. After the pilot study was completed, a random sample of 10 primary sources was extracted from the pool of primary sources that were not included in the pilot study. Then, a person independent of this study was instructed to perform structural and descriptive coding on the sample, using the code book that emerged from the pilot study. This number of sources was chosen due to the lack of available time of this second coder. At the same time, the researcher who conducted the pilot study applied the same procedure on the same sample. Then, Cohen's kappa [17] coefficient was computed. The magnitudes of this coefficient is used to assess the IRR and, therefore, to understand the degree of agreement between the two different coders. In general, a low magnitude reflects a low degree of agreement between coders. In this study, Cohen's kappa was interpreted following the guidelines of Landis and Koch [51]. In particular, a negative score is interpreted as no agreement at all, 0.01-0.2 is slight agreement, 0.21-0.4 is fair, 0.41-0.6 is moderate, 0.61-0.8 is substantial, and 0.81-1 is almost perfect. An agreement level below substantial shows that there could be significant differences in the interpretation of the theoretical framework applied in the coding process. Hence, a discussion of disagreements between the coders becomes of utmost necessity, in order to identify the root causes of these disagreements. In the context of this study, a discussion between the two coders existed whenever a slight or fair agreement was observed. The identified root causes have been addressed before proceeding to the full dataset coding phase.

The inter-rater reliability results are presented in Table 3.4. For RQ1, the agreement could be considered borderline substantial for both the structural and descriptive codes. However, for RQ2 there is an almost perfect agreement for the structural codes and a substantial agreement for the descriptive ones. Finally, for RQ3 there is a fair agreement for both structural and descriptive codes. This interpretation shows that although some degree of agreement exists for all research questions, some significant discrepancy exists in the analysis of the paragraphs relevant to RQ3 and RQ1.

	RQ1		**RQ2**		**RQ3**	
	Structural codes	Descriptive codes	Structural codes	Descriptive codes	Structural codes	Descriptive codes
Cohen's kappa	0.61	0.61	0.89	0.77	0.39	0.37

Table 3.4: The magnitudes for Cohen's kappa coefficient

The discussion revealed a few main issues, the most common being that some of the inferences performed by second coder went beyond the information presented in the text. This issue is due to the fact that the coder was not familiar with qualitative research, but also with the topic. A second issue is that some paragraphs were coded as "N/A", when they were, in fact, relevant. There are two main causes for this issue: lack of attention (in one case) and the second coder's lack of familiarity with the code book. When it comes to the low IRR scores for RQ3, the results are surprising. Despite the low magnitudes, the observed agreement was, in fact, 95%. However, Cohen's kappa is calculated based on both observed agreement and chance agreement. In the case of RQ3 coding, most of the paragraphs were coded as "N/A" by both coders. Hence, since there are very few examples where both coders agree on codes other than "N/A", the chance agreement becomes very high, which results in lower values for kappa. During the discussion for RQ3, we found out that the disagreements were caused by significant misunderstandings related to the description of codes regarding the challenges (in particular, the difference between cultural and social challenges).

3.6 Discussion

In this section, a discussion of the study findings will be provided.

The NoOps definition and positioning

The findings presented in the previous section show that there is no consensus regarding the definition of NoOps. The most frequently reported definition is that NoOps means developers do not have to interact with the operations team in their daily work. This elimination of interaction is supported by another common perspective, namely that NoOps means the complete exclusion of dedicated operations teams from the organization. A considerable number of perspectives suggest that a key component of the NoOps definition is the high level of automation in operations and the abstraction from the underlying infrastructure. However, although automation seems to be the common ground, different viewpoints exist regarding the utility of a dedicated operations team in this context of high automation - although some perspectives suggest that operations teams are still needed, others imply that automation renders dedicated operations teams unnecessary. Other perspectives suggest that NoOps is rather the goal or mindset of achieving complete automation of operations, such that no traditional operational tasks would have to be performed anymore. The definition of NoOps as being a goal or mindset is actually noteworthy, since this shows that NoOps is an ideal, rather than a set of practices. Hence, the meaning of NoOps is relative - while some public discourse participants define it from a technical perspective, others describe it from an idealistic standpoint.

The main components of the NoOps definition are the following:

- NoOps is a goal or mindset;
- High level of automation and abstraction from the underlying infrastructure;
- Traditional operations tasks are not performed anymore;

- Development and operations do not need to interact;
- The need for a dedicated operations team is reduced (or none);

Following the example of Jabbari et al. [42], we provide a definition that encompasses all the components enumerated above:

NoOps is the goal of having a highly automated IT environment, which is abstracted away from the underlying infrastructure, such that traditional operational tasks are not performed anymore, developers and operations teams do not need to interact and the need for a dedicated operations team is reduced or completely eliminated.

When it comes to the positioning of NoOps in relation to similar methods, there is absolute consensus: NoOps is an augmentation of DevOps. This fact has two main implications. First, implementing NoOps would be impossible if DevOps is not already adopted. Second, the augmentative nature of NoOps places the concept next to other DevOps augmentations, such as DevSecOps, DataOps, or BizDevOps [45].

The NoOps advantages

The adoption of the NoOps operational model seems to bring a considerable number of benefits, the most commonly reported being the social ones. In this regard, NoOps is expected to eliminate the friction between developers and infrastructure operations, thus allowing developers to focus on creating value, rather than spending their time and energy on operational aspects [R13, R28, R30]. The adoption of NoOps also provides benefits for the operations teams, as they have more time and opportunities for learning and growth, considering that a large part of their work is automated. Some of the less frequently reported social benefits, however, are increased accountability and improved collaboration (within teams).

The second most commonly reported type of advantages are those related to operational processes. In this area, NoOps is expected to increase the efficiency of operational processes, especially in terms of maintenance time [R76], deployment frequency [R59], or lead time [R14, R66]. Beside this higher operational efficiency, NoOps brings the advantage that the risk of human error is significantly diminished, since most operational processes become automated [R62]. Automation also brings the advantage of enhanced reliability [R68]. Another major benefit is that NoOps enhances operational processes with further intelligence, reducing the need for incident responses. This benefit is strongly linked to the use of proactive monitoring, which is defined [52] as continuously searching for indications that issues related to the system could arise, and flagging them for further investigation. Self-service is another benefit of NoOps due to the use of self-service portals for operations [R11]. With self-service, developers can perform operational activities (e.g., deployment) without having to delegate these tasks to a dedicated operations team [R6]. Finally, NoOps is expected to lower the risk for cyberattacks, provided that security protocol in a Serverless environment are done right, as mentioned in [R8]. However, the source does not describe what "done right" means in this context.

Another category of NoOps advantages are those related to the organization and business. The most commonly reported advantage is that of cost savings for infrastructure and application operations. These cost savings are strongly linked to the use of Serverless and pay-as-you-go computing solutions [R87, R96, R6, R13]. Due to these solutions, organizations do not have to invest in the acquisition and maintenance of on-premise or bare metal solutions [R30]. In addition to this, dedicated operations teams are not needed anymore, since Serverless solutions are managed by the offering vendor, so the end user is not required to deal with operations [R13, R30]. Another benefit of NoOps is that it builds in governance, quality and and compliance [R11].

Technical advantages are the least reported ones. The main technical advantage is that NoOps provides scalability, which can be explained by the use of Serverless solutions, which are designed to perform automatic scaling [R65, R87, R94]. In addition, NoOps encourages developers to write higher quality code [R29], and facilitates quality engineering [R4, R7] and test automation [R4]. These benefits are motivated by the fact that developers in a NoOps environment are often required to ensure that the developed system behaves the same in any environment [R16].

The characteristics of NoOps

The most commonly reported NoOps technical characteristics are those related to specific practices. While some practices are rather general (e.g., quality engineering or continuous delivery), others are described as specific to NoOps. The most prevalent technical characteristic in the analyzed literature is the outsourcing of operational activities to serverless or cloud service providers. As previously discussed, the use of serverless or cloud services comes with the advantage that the operational overhead is eliminated, since the customers do not need to perform operational activities, these being managed directly by the vendor [R87, R96, R6, R13]. Another commonly reported characteristic is the use of Everything-as-Code. According to an article of Escobar [24], Everything-as-Code involves the use of Infrastructure-as-Code, configuration management, containerization and orchestration. Another interesting characteristic is that of automation of operations based on artificial intelligence or machine learning. Solutions based on artificial intelligence could play an important role in proactive monitoring [R15], which involves the continuous searching for signs that a fault could occur in the future [52], but also in the development of self-healing software mechanisms [R94]. Self-healing represents the capability of software to detect its problems (through self-diagnostics) and fix them automatically [78]. Finally, chaos engineering is a characteristic recommended in the initial phases of NoOps adoption [R30]. It involves testing a system in production with the purpose of building "confidence in the system's capability of withstanding turbulent conditions" [1]. In other words, chaos engineering implies the introduction of random events that affect the system (e.g., a server failing), in order to understand the system behavior.

Other popular NoOps technical characteristics are related to operational aspects that can be automated. In this regard, the operational activities that could be automated under NoOps are monitoring [R10], infrastructure provisioning [R13], scaling [R34], testing [R2], reporting of critical information [R3], artifact building [R23], security scans [R27], deployment [R6], as well as the management of applications [R16]. There are also sources that do not mention individual aspects, but suggest the complete automation of test-release-operate phases of the software development lifecycle [R14, R42, R73].

From an architectural perspective, NoOps requires a microservice architecture. Since a microservice architecture enforces the separation of functional components into individual, loosely-coupled services, it has the important advantage that each service can be operated independently of the others, hence facilitating the integration with cloud services [R67, R14, R33].

Regarding NoOps-related tooling, the analyzed literature indicated the use of self-service portals for operations, allowing developers to perform operational tasks themselves, eliminating the interaction with the operations team [R6, R14, R38, R56]. Container orchestration mechanisms are also an integral part of NoOps, playing an important role in the automatic organization of various system components, such that the system behaves smoothly in production [R78, R94]. Finally, Continuous Integration tooling, software bots and virtual assistants are also suggested as operations automation solutions.

The most commonly reported cultural characteristics of NoOps are those of a collaborative nature. Perhaps the most important characteristic is that NoOps implies the removal of the silo between the development and operations teams. This is a rather surprising characteristic, considering the previously discussed fact that NoOps implies the automation of operational activities, from which it can be inferred that the gap between development and operations becomes even wider. However, the analyzed literature suggests that especially in the adoption phases of NoOps, a strong collaboration between development and operations is necessary, since together both teams could devise a proper automation strategy [R9, R95]. Moreover, this collaboration also leads to a diffusion of responsibilities, since developers become involved with operations, while operations people get involved in development [R6, R9, R10, R11, R14, R20, R82]. Hence, a transfer of skills occurs between the two teams. NoOps is also expected to remove the walls between the business and the IT teams.

Responsibility shifting in NoOps is done towards the left of the software development lifecycle. In other words, the responsibility for deployment, security, monitoring, quality assurance and operation become the responsibility of developers, rather than that of dedicated operations teams.

From an organizational perspective, the adoption of NoOps requires training teams on new automation tools, as well as the involvement and acceptance of the management. Moreover, from a mindset perspective, developers are expected to take ownership of the developed software and aim for process reproducibility, in order to ensure a facile integration of automation solutions [R2]. Finally, NoOps requires proactivity in engineering the variability of operations. As discussed in [R13], the adoption of NoOps leads to a significant change in the mindset of operations people, since automation of operations causes them to redefine their role. As the author highlights, instead of performing operational activities from a plumber's perspective (fixing issues after they happen), NoOps provides operations staff with the opportunity to approach their job from an engineering perspective (e.g., by automating their tasks).

The challenges of NoOps

The most commonly reported challenges of NoOps are those of a technical nature. The most prevalent technical challenge is that NoOps might not be suitable for any kind of system, requiring the re-engineering of legacy ones. As previously discussed, the adoption of NoOps has a number of requirements, both of technical and cultural essence. For example, microservice architecture is a necessity, since its modular and loosely-coupled nature enables a facile integration with cloud services. In contrast, a complex system with a monolith architecture can hardly be integrated with cloud services for multiple reasons, including lack of fault tolerance, long deployment times and inability of deploying individual components [74]. Due to these disadvantages, the automation of operational activities related to a monolith system is hardly possible. Another common challenge of NoOps is the fact that outsourcing activities to third-parties could lead to technical issues and limitations. Since NoOps is strongly tied to the use of third-party cloud services and considering the fact that the operations are completely outsourced to the cloud vendors, this means that if an unfortunate technical event occurs on the premises of the cloud vendor, the availability of the system using those services is also affected, without any possibility to intervene [R37]. Another example is that the users of cloud services have to rely strictly on the out-of-the-box functionality of these services, without any possibility to customize the internal mechanisms. For instance, users do not have the possibility of customizing load balancing algorithms or auto-scaling processes [R62, R37, R13]. Another important NoOps technical challenge is that of vendor lock-in, which is defined as the "expensive and time-consuming migration of application and data to alternative providers" [67]. Beside the technical limitations and tight coupling to cloud vendors, another challenge is that a low-quality NoOps implementation could lead to a high risk of cyberattacks. For instance, the media reported cases [84] where improper security configurations for certain cloud services led to major privacy and security breaches. Finally, the last technical challenge is the management of functions when using Function-as-a-Service solutions. Function-as-a-Service is a Serverless programming model in which the developer defines the application as a set of ephemeral, stateless and containerized functions which are executed using the resources of a cloud service provider in response to certain events [81]. Due to the small size of these functions, defining an application as a set of functions results in a large number of deployment units [R17]. Hence, the management of an increasing number of deployment units becomes a high priority issue.

From a cultural perspective, NoOps has major challenge, namely the fact that it requires a mindset shift for developers in order to ensure that the developed artifact behaves the same in both development and production environments. Often, developers fall into the trap of believing that their code has the same behavior in any environment, which is not always the case [R16, R29, R78, R86]. Hence, overcoming this pitfall requires a significant mindset shift.

The social dimension of NoOps also brings a few challenges. The most commonly reported one is that NoOps causes discomfort in the initial phases of adoption. The causes for this discomfort include the fact that developers have more responsibilities [R9, R85], the fear of operations staff of losing their job [R88], as well as the difficulties in adapting to a new way of working [R1]. Moreover, teams are also hesitant to understand the business requirements in the initial phases of adoption [R9]. Since NoOps requires a dramatic shift to automation solutions, the NoOps transition teams are often required to adhere to specific budget requirements from the business (e.g., financial), leading to significant constraints in the early decision making processes. Finally, one source reported that NoOps can not be based on traditional ways of working together [R9]. However, the source does not explain exactly why this is the case.

NoOps is also associated with one organizational challenge, namely that the adoption of NoOps requires addressing regulatory compliance, security and other cross-cutting concerns company-wide [R11]. This could become problematic for many organizations, especially for those that process very sensitive data. If such an organization decides to use the services of cloud vendors, then the transfer of sensitive data becomes subject to certain regulatory frameworks, which could lead to severe consequences in case of violations [R24].

Lastly, one general challenge reported in the literature is that adopting NoOps could lead to less involvement from development in operations, which completely contradicts DevOps practices. As previously discussed, NoOps implies a diffusion of responsibility among development and operations teams, as well as transfer of skills. In addition, NoOps is an augmentation of DevOps, which means that adopting NoOps would be impossible without a proper implementation of DevOps. Hence, NoOps does not represent the end of DevOps, but rather its natural evolution. In NoOps, there are no development and operations teams anymore, but rather a single team that works for the end goal of creating value for the business [R39].

The representation of NoOps in practitioner's literature

In section 3.4.3, a number of heatmaps were introduced, highlighting the mapping between the findings and the types of organizations reflected in the primary sources.

Looking at the heatmap depicted in Figure 3.12, it is obvious that most of the advantages are reported by product development and consultancy companies. Hence, one could wonder whether these advantages are actually based on solid evidence or are just part of a marketing strategy, aiming at promoting the products or services offered by these companies in order to attract potential clients.

To understand whether the claims of these companies are true, they need to be compared and contrasted with an additional viewpoint, which is that of practitioners. The practitioners' viewpoint will be analyzed in the next chapter.

3.7 Threats to validity

This section introduces a number of threats that could affect the validity of the results presented in this study, based on the validity criteria of Wohlin et al. [91].

Construct validity

Construct validity refers to whether the theoretical constructs in the study are correctly represented and interpreted in the operational phases [91] (i.e., whether the measured variables "correspond to the intended meanings of the theoretical terms" [22]).

One threat that affects both construct and external validity is that the search process might have not included all the synonyms for the main keywords in the research questions. Hence, the search process might not be thorough enough, so potentially relevant sources could have been missed. Another search limitation is the fact that the search was performed only on a limited number of search engines and only English results have been considered.

A second threat is represented by the lack of evidence in some of the selected primary sources. The lack of evidence in some sources causes an epistemological issue, since "we do not know, how we know, what we know" [31]. In addition to this, since many of the sources have been released by companies, there could be a possibility that some results might be biased.

Another threat is that the first research question refers to how NoOps is described in practitioners' literature. What a description of a concept represents, however, could be considered subjective. While in this study "description" refers to NoOps definition, advantages and positioning, others could interpret it differently (e.g., historical context or practitioners' opinion, among others).

Internal validity

Internal validity refers to whether the relationships between the treatment and the outcome are justified, and are not the result of a neglected factor [91]. For systematic literature studies, internal validity is related more to the validity of the research methods used in the study.

The main threat to internal validity is related to the subjectivity biases. Since this study is of a qualitative nature, employing content analysis methods in order to derive the findings, there is a risk that reasoning errors might have been made in the process. In addition, the findings have not been validated by a domain expert. Hence, the results of this study could be considered merely a reflection of the author's thoughts and beliefs, especially since the entire study was conducted by a single person. Subjectivity biases could have been introduced in multiple stages, including the sample selection, quality control, as well as data extraction and analysis. To mitigate these biases, inter-rater reliability assessment has been conducted for the analysis of a sample of primary sources in order to measure the degree of subjectivity and fine-tune the analysis process. However, the conducted IRR assessment has some limitations, the first being that the second coder did not have a proper understanding of qualitative research, as well as enough knowledge about the studied topic. Therefore, the utility of this assessment is questionable, since it is unclear whether the agreement reached after the discussion is actually unbiased. A second limitation of the IRR assessment is the fact that it was conducted on a small random sample, which might not be representative for the entire dataset. Finally, in order to compensate for the lack of validation from a domain expert, we conduct a complementary study of NoOps-related online discussions, such that the practitioners' viewpoint is also considered. This complementary study is discussed in chapter 4.

External validity

External validity refers to whether the results of this study can be generalized to a wider context [91]. In this study, there are two main threats to external validity.

First of all, one concern is the fact that the analysis was conducted based on the results of a pilot study, for which theoretical saturation might not have been reached (i.e., the code book obtained from the pilot study might not have been exhaustive). Due to the possible lack of theoretical saturation, the findings of the study might not depict a full-scale picture of NoOps. To mitigate this threat, in the full dataset coding phase we accounted for this possibility, marking potentially relevant new codes as "Other". However, no other codes were discovered in the end, which proves that the results of the pilot study provided an accurate reflection of the entire dataset. Another concern related to the pilot study is the fact that the "chunked transcription" method proposed by Christian Bird [10] was used before the analysis of audio and video sources. Hence, there could be a chance that potentially relevant findings might have been lost during this transcription process. To limit the impact of this threat, the transcriptions have been double-checked before the analysis was conducted.

A second threat to external validity is the fact that relevant literature might have been missed during the literature search process, since not all possible synonyms related to the main keywords in our research questions have been used. If relevant literature was missed, there is a risk that the study findings might not be comprehensive. Moreover, the search process has been conducted only once, in December 2020, so the findings might be representative only for a specific time-frame. To limit the impact of this threat, one possibility would be to replicate the search process in the future, in order to identify new primary sources and enhance the study findings.

Conclusion validity

Conclusion validity refers to whether there is a valid relationship between the treatment and the outcome [91], i.e., whether the conclusions derived from the study are actually emerging from the data.

The main threat is that the conclusions in this study are drawn by a single author, hence interpretation bias could exist. However, to partially mitigate this threat and improve the reproducibility of this study, a replication package has been created.[2]

Reflexivity statement

As the sole researcher involved in this qualitative study, I am aware of the impact that my own beliefs and experiences can have on the validity and completeness of the results. Hence, with this reflexivity statement, I want to provide a brief clarification of my ontological and epistemological positioning in relation to the current study.

Let me start by introducing some context for this work. I started working on this topic due to my curiosity regarding IT operations and automation. Since I have always been involved in activities regarding the left-side of the software development lifecycle, the area of operations has always been a mystery for me. However, the idea of automating operations is an even bigger mystery. Hence, when I started working on this topic, I only had a minimum understanding of what operations entail. Moreover, I have never been part of a DevOps team, my knowledge being limited only to basic information I gathered out of curiosity. When I started this work, I had two main goals. The first one was that I wanted to learn as much as possible about the area of NoOps, while the second one was that I wanted to contribute towards bridging the gap between the industrial practice and academic research. Having been involved in industrial activities for a few years, I can clearly see the gap between what is happening in the industry and research.

My industrial experience could also impact the results of this study. As it could be seen throughout this chapter, the focus was more towards the technicalities discussed in the literature, rather than towards evaluating the presence of evidence for strong claims, such as reported advantages. While this could be due to my lack of experience with qualitative research, it could also be due to the fact that during the analysis I always attempted to relate some of the findings to my industrial experience. For example, the fact that I have been a long-time user of cloud services might have guided my attention more towards cloud-related information. In other words, I believe that my industrial experience could have had an impact on the analysis and interpretation of the data, fact that strengthens again the prevalence of subjectivity in this study.

Finally, I believe that the philosophical stance that best represents the work conducted in this entire study is pragmatism. As discussed by Easterbrook et al. [22], pragmatism involves acknowledging that all knowledge is approximate, being judged by its usefulness for addressing practical problems. Given the relativistic nature of pragmatism, pragmatists highlight the idea of consensus, favoring mixed-methods research in order to understand the truth.

[2]Link to replication package: `https://github.com/drgs/noops-slm-reproc`

Chapter 4

NoOps in online discussions

The goal of the fourth research question is to capture a snapshot of the practitioners' viewpoint by analyzing NoOps-related online discussions. Although the previous chapter provided an analysis of NoOps-related gray literature, a significant part of this literature was published by companies, but also by publishers with a low credibility and outlet control, such as blogs. Hence, it could be that some of the insights gathered from practitioners' literature might not be accurately representing practitioners' point of view. To overcome this limitation, we follow the recommendations of Barik et al. [9] of complementing the existing findings with knowledge from online discussions from social news websites. According to Easterbrook et al. [22], this approach of using different methods and data sources concurrently in order to confirm or cross-validate findings is called concurrent triangulation.

Social news websites [89] are a form of user-maintained online news aggregators. On these platforms, users can start discussions about certain topics and post references to articles that might be interesting for the community. This type of contributions are called discussion threads. These discussion threads are usually ranked by their popularity, which is defined based on the votes of other users. In addition, users can express their thoughts regarding the topic by posting comments on these threads.

Due to the fast-paced nature of software engineering, practitioners often need to stay up-to-date with the most recent news and approaches in the field. The most common examples of social news websites among developers are Reddit[1] and HackerNews[2] [4]. Hence, in this study we collect data from these two platforms, considering their high popularity among developers [4, 9].

An overview of the process used to conduct the analysis of online discussion is illustrated in Figure 4.1. The process begins with the search for discussion threads on both Reddit and HackerNews. Then, the search results are filtered manually based on their relevance to NoOps. After this filtering step, all comments from the selected discussion threads are extracted. After comment extraction, topic modeling methods are applied in order to synthesize the content of these comments, the result being a number of discussion topics. These topics are then compared with the findings from the qualitative study, which were discussed in the previous chapter. This way, it can be investigated which qualitative findings are supported by practitioners' viewpoint. However, topic modeling has one main limitation, namely that the resulting discussion topics provide a high-level synthesis of the content of the analyzed data. This means that important information, such as the attitude of practitioners in those comments, is not captured. In other words, although it can be inferred what is being discussed (topics), it is difficult to understand how things are being discussed (attitudes). Hence, opinion mining is performed in order to overcome this limitation and draw a clear picture of the practitioners' viewpoint by extracting the opinionated comments. These opinionated comments are then subject to manual closed coding, and the findings are then compared with the perspectives reflected in practitioners' literature.

[1] https://www.reddit.com
[2] https://news.ycombinator.com/

This chapter is structured as follows. Section 4.1 introduces the data collection process and results. Then, section 4.2 describes the approach that is used to synthesize and interpret the data, which is based on topic modeling. Section 4.3 presents the opinion mining approach, which is used to improve the picture of practitioners' attitude towards NoOps-related concepts in online discussions. Then, section 4.4 provides a deeper discussion of the findings presented in this chapter, while section 4.5 introduces the threats to validity associated with the presented work.

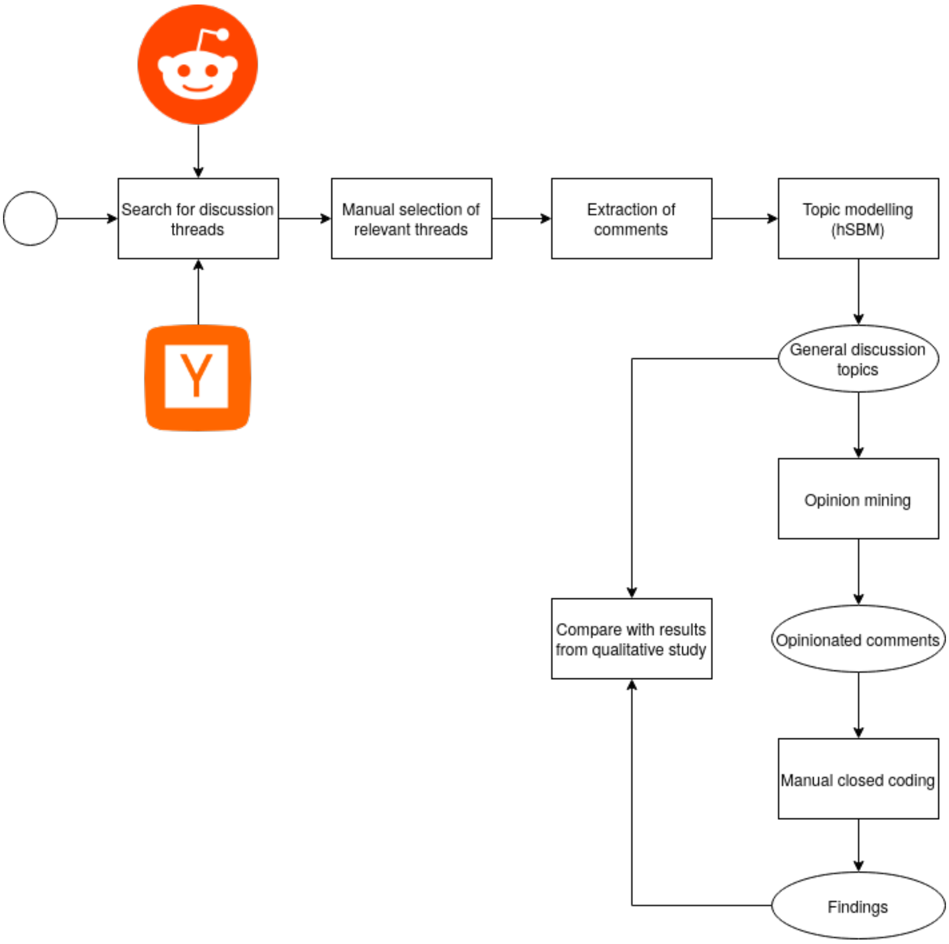


Figure 4.1: The general workflow for conducting the analysis of NoOps-related online discussions

4.1 Data collection

This section introduces the process for collecting the online discussion data from the social news platforms, as well as the data collection results.

4.1.1 Data search and selection

The data collection process was comprised of the following steps. The first step involved searching for all possible NoOps-related comments on both Reddit and HackerNews. The following search strings have been used for both platforms, taking into account the fact that the search functionality on both platforms is exhaustive:

NoOps OR No-Ops OR No Ops OR "NoOps" OR "No-Ops" OR "No Ops"

In order to perform the search on Reddit, the PRAW library[3] is used, which is a Python-

[3]https://praw.readthedocs.io/en/latest/

based wrapper for the Reddit API. We decided to use PRAW because it is designed in a way that overcomes some limitations of the Reddit API (e.g., search rate limits) and simplifies the search process. The search is performed on the thread title and body, with no temporal constraints (i.e., all search results have been extracted, without accounting for the submission time). The only employed search constraint is that the number of comments in a thread is greater than 0, since a thread with no comments does not provide any useful insights.

The search on HackerNews was performed using the Algolia HackerNews Search API[4]. The search was conducted using a number of constraints, including the number of comments being greater than 0 and strict typo tolerance, which means that only the results that precisely matched the search strings were returned. It is important to note that the HackerNews API enables searching for both discussion threads (thread title and body) and comments (comment body), unlike Reddit which supports only searching on discussion threads. Hence, for HackerNews the search is conducted on both threads and comments.

When the search for discussion threads on both Reddit and HackerNews was completed, the next step was to remove the duplicated results obtained on each platform, based on the thread ID. Then, the results for both platforms were merged together.

The next step was to perform a manual filtering on the results based on their relevance to the NoOps concept, forming a pool of relevant discussion threads. The relevance was assessed based on the title and body of the discussion thread, based on the following framework. A discussion thread was marked as relevant if at least one of the following was true:

- It contained words related to IT operations;
- It contained the word NoOps/No Ops/No-Ops, as well as words related to software development;
- It contained the word NoOps/No Ops/No-Ops, as well as names of (cloud) vendors offering products or services related to at least one of the technical characteristics identified in the previous question;
- It contained the word NoOps/No Ops/No-Ops, as well as names of tools that can be used for the management of cloud-related products or services.

The same filtering was conducted for the HackerNews search on comments - all comments that contained information related to NoOps were included (following the previously discussed framework), while the others were discarded. Then, for each remaining comment, its parent discussion thread was extracted and added to the pool of relevant discussion threads.

After obtaining the final pool of relevant discussion threads, all comments that are part of these threads are extracted into a CSV file. These comments represent the final dataset used in the analysis.

The entire search process was conducted on April 9, 2021. All search results can be found within the replication package.[5]

4.1.2 Data collection results

Table 4.1 shows the search results statistics for Reddit. Overall, 537 unique threads were identified. After the manual filtering, only 9 of those threads remained. This sharp reduction is caused by the high number of results that were not relevant to NoOps. It was surprising to find out that the term NoOps is used not only in the software engineering community, but in many others (e.g., gaming or gender-related communities). From these 9 threads, 57 comments were extracted.

[4] `https://hn.algolia.com/api`
[5] Link to replication package: `https://github.com/drgs/noops-slm-reproc`

Search string	**# results (threads)**
NoOps	244
No-Ops	237
No Ops	239
“NoOps”	244
“No-Ops”	240
“No Ops”	241
Total	
with duplicates	1445
without duplicates	537

Table 4.1: Reddit search results

Table 4.2 shows the search results statistics for HackerNews. 176 unique threads were identified. After the manual filtering of the threads, only 7 threads remained. From these 7 threads, 161 comments were extracted.

The search for comments resulted in 2840 results. The manual filtering of these comments resulted in 82 NoOps-related comments, which originate from 64 different threads. From these additional 64 threads, 8129 comments were extracted, leading to a total of 8290 comments extracted only from HackerNews.

Search string	**# results (threads)**	**# results (comments)**
NoOps	148	2160
No-Ops	129	2044
No Ops	197	3033
“NoOps”	2	127
“No-Ops”	9	252
“No Ops”	9	252
Total		
with duplicates	494	7868
without duplicates	176	2840

Table 4.2: HackerNews search results

Combining the Reddit and HackerNews search results, the entire dataset consists of 8347 comments. Considering that these comments have a cummulated length of 450000 words, the manual analysis of this comments would take a very long time, which would not be feasible considering the time constraints associated with the current thesis project. Hence, the decision of using automated analysis methods was made.

4.2 Topic modeling of online discussions

As discussed in the previous section, due to the large number of comments, manual analysis requires a considerable amount of time. Hence, automated analysis methods are required. The goal of the current research question is to understand how NoOps is described in online discussions, so that the differences between the practitioners' viewpoint and the findings from literature can be compared. Hence, to answer this research question it is necessary to extract the key concepts that are discussed in the entire dataset (i.e., a task that is similar to the manual qualitative analysis conducted in the previous chapter). The approach used to perform this task is called topic modeling [11].

Topic modeling has been used in a variety of fields, such as software engineering [50], security [15, 16], social sciences [55] or medicine [57]. Similar to the current study, there are many papers that applied topic modeling methods to analyze online discussions on Reddit [43, 32, 59], but also on social media platforms such as Twitter [38].

The rest of this section is structured as follows. Subsection 4.2.1 introduces some topic modeling approaches and discusses the approach used in this study, together with the data preprocessing steps. Moreover, subsection 4.2.2 provides an overview of the topic modeling findings.

4.2.1 The topic modeling approach

Latent Dirichlet Allocation

The most popular approach [88] for the topic modeling of unstructured texts is the Latent Dirichlet Allocation (LDA), proposed by Blei et al. [12]. The main idea behind LDA is that each document in the dataset is represented as a random mixture of topics, whereas each topic is represented as a probability distribution over words. The main assumption behind LDA is that the distribution of topics over all documents, as well as the distribution of words over topics, follow a Dirichlet distribution. It is important to note that in LDA, the number of topics has to be known beforehand, making it a parametric approach. Moreover, the underlying Dirichlet distribution requires the configuration of two hyperparameters (both of scalar nature): α and β. A high α value results in documents that are represented by a high number of topics, while a low value leads to documents that are represented by a lower number of topics. Similarly, β reflects the mixture of words within a topic - a high value leads to topics that contain a wide variety of words, while a low value results in topics consisting of fewer words. As discussed in the paper of Annibale Panichella [68], many studies that used LDA without performing hyperparameter tuning faced sub-optimal results, emphasizing the importance of choosing the right parameters for a dataset. The study presents a comparison between multiple meta-heuristics for LDA hyperparameter tuning, applied on the software engineering task of duplicate bug report identification. The study concludes that there is no meta-heuristic that outperforms the others, which implies that there is no right way of performing LDA hyperparameter tuning, despite its high importance.

The chosen approach

In another paper, Gerlach et al. [33] discuss the fundamental flaws of LDA, including the inability of choosing the right number of topics without prior knowledge of the dataset, the difficulties of hyperparameter tuning, the lack of justification of Blei et al. [12] for the choice of Dirichlet priors, as well as the model's incompatibility with statistical properties of natural text. To address these limitations, Gerlach et al. [33] proposed a state-of-the-art topic modeling approach based on the problem of finding communities in complex networks.[6] In this approach, the text corpus is represented as a bipartite graph of documents and words, the weight of the edges being represented by the frequency of each word in a document. Then, a community detection method called hierarchical stochastic block modeling with non-parametric priors is applied on this representation.[7] This method makes fewer assumptions about the structure of the data, in contrast to LDA, due to the non-parametric priors. The evaluation results show that this approach outperforms LDA on both real and synthetic corpora of different lengths, and that it successfully addresses the limitations of LDA by automatically inferring the number of topics, and by providing a higher compatibility with the statistical properties of natural texts. Due to these advantages, as well as due to the non-parametric nature of the approach, we decided to use this method to perform the current topic modeling tasks.

[6] In the paper, the authors show that the problem of community detection is mathematically equivalent to the problem of topic modeling, subject to certain constraints on the network structure

[7] As discussed by Gerlach et al. [33], the stochastic block model is a mathematical generalization of LDA's generative basis

Interpretation of topics

Since the resulting topics are just a collection of words, their interpretation needs to be performed manually. To facilitate the interpretation, a word cloud visualization [5] is used, which is based on the 30 most frequent words in each topic. This type of visualization is used to display the most frequent words in a topic, the size of each word being proportional to its frequency. After the interpretation, the topics are compared with the qualitative literature findings through a mapping between topics and findings.

Data preprocessing

The data preprocessing phase consisted of multiple steps that were inspired from literature [36, 15, 68, 37], as enumerated below:

1. **Dataset cleaning:** This step consists of removal of HTML entities, emojis, numbers, punctuation marks, code snippets and hyperlinks. The reason for removing these items is the fact that they do not bring any semantic value to the topic model. Then, all words were converted to lowercase.

2. **Tokenization:** Every comment in the dataset is converted into a list representation consisting of the words associated with that comment (called tokens). This step is a necessary precondition of the selected topic modeling approach.

3. **Removal of stop words:** Natural texts contain words that are very common, but do not bring any significant information (e.g., "and", "so", "this", etc). These words are called "stop words". Since these words are used in almost every comment, keeping them would pollute the resulting topic model. Hence, they are removed from each comment using the standard stop words list for English provided by NLTK.[8]

4. **Lemmatization based on part-of-speech tagging:** Lemmatization involves generating the inflected form of a word, based on its morphological information (obtained via part-of-speech tagging). For example, words such as "going" and "went" are reduced to the same form, which is "go". Hence, lemmatization helps with reducing the number of words in the corpus, while preserving the semantic value. Similar to the work of Guzman et al. [36], we use the WordNet lemmatizer, proposed by Miller [62].

5. **Identification of collocations:** In natural texts, words that frequently co-occur in a given context can often be observed (e.g., "United States" or "AWS Lambda"[9]). Hence, this step involved identifying the collocations by generating 2-shingles and 3-shingles, which represent sequences of two or three adjacent words in a string [13]. To detect whether a shingle is a collocation, the NLTK library was used.

6. **Removal of words that occur only once in the dataset:** As previously discussed, the chosen topic modeling approach is based on the representation of the entire corpus as a bipartite graph, with weighted edges between the comments and their associated words. Since the stochastic block model determines the communities (topics) mainly based on the edge weights (denoting the frequency of a word in a comment), this means that nodes that have only one edge with a weight of 1 do not have any significant impact on the generated model. Hence, removing these edges helps in reducing the computational power required for fitting the model, since the network becomes lighter.

[8] `https://www.nltk.org/`
[9] `https://aws.amazon.com/lambda/`

4.2.2 Findings

The application of the hierarchical stochastic block modeling approach of Gerlach et al. [33] resulted in 56 different topics. The word clouds used to interpret these topics can be found in Appendix F.

Out of the 56 different topics, only 17 were interpreted as being related to NoOps. Table 4.3 shows the interpretation of these NoOps-related topics, as well as their mapping to the findings that emerged from the literature study (as presented in Chapter 3.4). For completeness purposes, the interpretation of all 56 topics can be found in Appendix G.

As it can be observed, topics 23, 25 and 32 could not be mapped to any findings, despite the fact that they are related to NoOps. Moreover, it can be noticed that most topics contain clear references to specific cloud products and services, implying that the discussions might be focused on reviews and experiences related to certain products, rather than on high-level NoOps-related aspects. In contrast, most gray literature sources discussed high-level aspects, rather than specific products. This fact also implies that an accurate interpretation of the topics can only be achieved by taking into account the objectives and use cases of those specific tools, products and services.

Topic ID	Top-30 Words	Interpretation	Mapping to codes from literature study
1	microsoft, azure, tie, license, extension, incentive, entity, visual_studio, vendor_lock, slack, premise, portal, equal, mapping, mvp, jumping, really_hope, edition, sends, played, trap, burst, macos, inferior, error_handle, silver, pwas, visual_studio_litter_sql, throat, machinery	Microsoft Azure[10] is a cloud service provider. The challenge of vendor lock-in is also mentioned, showing that there was probably a discussion about the challenges of using Azure services.	Outsourcing operational activities to serverless/cloud providers, Vendor lock-in
9	heroku, paas, uptime, redundancy, elastic_beanstalk, appfog, openstack, dynos, cloud_foundry, manage_server, bash, beanstalk, firebase, monolith, openshift, dyno, non_trivial, autoscale, manages, pivotal, bosh, prototyping, dotcloud, buildpacks, vsphere, availability_zone, middle_ground, joyent, add_ons, git_push	Multiple cloud-related services and vendors are discussed (Heroku[11], Elastic Beanstalk[12], AppFog[13], OpenStack[14], CloudFoundry[15], Firebase[16], OpenShift[17]). Autoscaling is mentioned as well. We see some cloud-specific terms as well, such as dynos (Heroku-specific containers) and availability zones.	Outsourcing operational activities to serverless/cloud providers, Auto-scaling

[10] https://azure.microsoft.com/
[11] https://www.heroku.com/
[12] https://aws.amazon.com/elasticbeanstalk/
[13] https://github.com/appfog
[14] https://www.openstack.org/
[15] https://www.cloudfoundry.org/
[16] https://firebase.google.com/
[17] https://www.openshift.com/

11	use, user, add, file, access, api, serverless, nice, function, use_case, check, via, directly, easily, event, us, error, interface, implementation, remove, custom, feel_like, documentation, allows, automatically, response, integration, connect, pain, fairly	We see a clear mention of Serverless technologies, as well as several words regarding their usage and implementations, which clearly shows that the discussion is related to Serverless implementation models.	Outsourcing operational activities to serverless/cloud providers
13	server, instance, ec, cluster, setup, load_balancer, spin, provision, vps, ec_instance, downtime, dns, vpc, redis, load_balance, bare_metal, automatic, take_care, eb, flexible, digitalocean, failover, elastic, auto_scale, bucket, az, ip_address, rout, ridiculous, prem	Several AWS-specific concepts are mentioned, such as EC2 service instances[18], Virtual Private Clouds[19] and S3 buckets[20]. EB stands for Elastic Beanstalk, another AWS service. We also see mentions of load balancers and auto-scaling, but also the words "server" and "provisioning", which could indicate the use of automated infrastructure provisioning. Clearly related to NoOps.	Outsourcing operational activities to serverless/cloud providers, Auto-scaling, Automated infrastructure provisioning
17	aws, service, cloud, amazon, customer, offer, gcp, account, provider, offering, migrate, cloud_provider, pricing, migration, gce, shut, reliability, cloud_service, outage, beta, gc, aws_azure, manage_service, beat, rackspace, iaa, impression, commitment, compelling, saving	Topic is related to cloud vendors (AWS, Google Cloud Platform[21], Azure) and their provided services (e.g., Google Compute Engine[22]). Could also indicate technical challenges associated with the use of these services (due to words as "outage" and "reliability"), but there is not enough information to derive that this is indeed the case.	Outsourcing operational activities to serverless/cloud providers
18	test, deploy, production, environment, deployment, task, script, configuration, debug, deployed, prod, command, shell, ci, ssh, execute, deploys, session, puppet, git, directory, environment_variable, locally, terminal, stag, cli, fleet, repository, jenkins, repo	Related to automated deployment, CI tools (Jenkins[23]), Everything-as-Code (Puppet[24]) and Automated Testing	Automated deployment, Continuous Integration tooling, Everything-as-Code, Automated testing

[18] https://aws.amazon.com/ec2/
[19] https://aws.amazon.com/vpc/
[20] https://aws.amazon.com/s3/
[21] https://cloud.google.com/
[22] https://cloud.google.com/compute
[23] https://www.jenkins.io/
[24] https://puppet.com/

20	build, docker, image, layer, dependency, package, install, runtime, instal, compile, nginx, docker_image, docker_container, jar, binary, ubuntu, orchestration, final, dockerfile, frontend, plugin, maven, immutable, bundle, artifact, npm, solves, base_image, builder, horizontal_scale	Related to Docker[25] and container orchestration.	Orchestration mechanisms
23	cost, pay, month, free, expensive, hour, cheaper, save, internal, spend, vendor, spending, spent, dedicate, youre, virtual, afford, alone, scalability, mo, full_time, minimum, expertise, headache, fee, spend_time, save_money, good_luck, hobby, round_error	Seems to be related to costs associated to the use of services offered by vendors, but it is unclear whether this discussion is related to the "cost savings" benefit identified in literature or not, due to words such as as "expensive". Still, it can be related to NoOps, due to the presence of words such as "vendor".	N/A
25	bill, charge, capacity, billing, cap, credit, payment, discount, monthly, reserve, destroy, peak, reserve_instance, xlarge, throttle, flat_rate, threshold, hobbyist, harm, heavy_user, destroyed, hourly, physically, willing_pay, quota, separately, upfront, pricing_model, capacity_need, pay_per	Probably related to the costs incurred for hobby cloud applications. Since we see the expression "reserve instance", this could definitely be related to the usage of cloud services. However, it can not be mapped to a specific finding due to insufficient information.	N/A
28	kubernetes, gke, pod, eks, docker_compose, agent, kube, simplify, autopilot, nomad, docker_swarm, swarm, controller, persistent, kubernetes_cluster, learn_curve, joe, stateless, ingres, stateful, service_discovery, web_page, ww, overkill, dead_simple, kubectl, bob, rac, yml, boon	Mainly related to orchestration mechanisms: there are tools such as Kubernetes[26], Docker Compose and Docker Swarm, as well as Hashicorp Nomad[27]. There are also some vendor-provided orchestration services, such as GKE (Google Kubernetes Engine[28]) or EKS (AWS Elastic Kubernetes Service[29])	Orchestration mechanisms

[25]https://www.docker.com/
[26]https://kubernetes.io/
[27]https://www.nomadproject.io/
[28]https://cloud.google.com/kubernetes-engine
[29]https://aws.amazon.com/eks/

32	software, development, everyone, term, automate, focus, automation, role, programmer, engineering, skill, culture, fire, position, blame, deep, communication, theory, therefore, fight, wall, meaning, software_development, software_engineer, embed, direction, board, productive, responsibility, communicate	Clearly related to NoOps. There are words such as automation, communication, culture and responsibility, which commonly occurred throughout the literature. However, this topic can not be mapped to specific findings, due to the lack of specific information.	N/A
35	lambda, log, api_gateway, aws_lambda, lambda_function, mb, cold_start, dynamodb, execution, endpoint, cloudwatch, nodejs, alert, serverless_framework, side_project, take_second, zappa, resize, invocation, claudia, cloudwatch_log, cold_start_time, async, cron, response_time, gateway, azure_function, boto, web_interface, apex	The discussion seems to be related to serverless-related solutions, especially Function-as-a-Service offerings from various companies (e.g., AWS Lambda, Azure Functions). AWS API Gateway[30] is also mentioned, since it is often used together with Lambda. We also see services such as DynamoDB[31], which is the fully managed NoSQL database solution offered by AWS. AWS Cloudwatch[32] is also often used together with Lambda for monitoring purposes, which indicates the use of Automated Monitoring. This topic clearly indicates a relation to NoOps, especially the outsourcing of operations to cloud vendors.	Outsourcing operational activities to serverless/cloud providers, Automated monitoring
37	store, key, secret, password, ansible, vault, credential, encrypt, encryption, iam, token, hash, auth, hashicorp, audit, km, temporary, username, submit, playbook, metadata, decrypt, tunnel, hashicorp_vault, secret_management, regulatory, user_id, iam_role, truncate, env_var	Related to Infrastructure-as-Code (Ansible[33]), as well as the management of secrets (words such as store, IAM, encryption, Hashicorp Vault[34], credential, secret_management)	Everything-as-Code

[30] https://aws.amazon.com/api-gateway/
[31] https://aws.amazon.com/dynamodb/
[32] https://aws.amazon.com/cloudwatch/
[33] https://www.ansible.com/
[34] https://www.vaultproject.io/

47	devops, team, ops, operation, engineer, hire, devs, sysadmin, organization, noops, sysadmins, title, staff, admin, ops_team, devops_engineer, infra, agile, department, empathy, silo, ops_people, devops_team, small_company, dev_ops, specialist, principle, qa, mindset, dev_team	Interesting enough, the concepts DevOps and NoOps are mentioned, as well as concepts related to operations teams. The word "silo" could refer to the traditional separation between development and operations ("devs" and "ops").	Removing the silo between Dev and ITOps teams
48	google, google_cloud, twitter, spotify, partner, music, big_data, acquisition, gmail, cloud_platform, customer_support, disclosure_work, data_studio, bi, announcement, loud, bigtable, looker, reputation, sm, paid_support, notify, engagement, partnership, hangout, play_store, web_ui, dataproc, case_study, live_migration	Clearly related to the services offered by Google Cloud, and how Spotify (which is a client of Google Cloud Platform[35]) uses these services.	Outsourcing operational activities to serverless/cloud providers
52	terraform, queue, cloudformation, fargate, template, sqs, elb, convox, cf, ami, kinesis, autoscaling, consul, use_terraform, security_group, cloudfoundry, aws_fargate, hcl, autoscaling_group, kinesis_stream, cloudformation_template, reusable, trigger_lambda, codebuild, cf_template, asg, logstash, sn_sqs, scale_horizontally, fud	The words "Terraform"[36], "HCL" (Hashicorp Configuration Language) and "CloudFormation"[37] suggest the use of Infrastructure-as-Code. There are also some PaaS solutions mentioned, such as CloudFoundry and Convox[38]. AWS Fargate[39] is a Serverless compute engine, which completely abstracts away infrastructure details. AWS Kinesis[40] enables distributed data processing of streaming data. SQS[41] is another AWS service which provides message queueing services. Finally, there are some mentions of AWS Lambda, as well as auto-scaling. Clearly related to NoOps.	Outsourcing operational activities to serverless/cloud providers, Everything-as-Code, Auto-scaling

[35] https://cloud.google.com/customers/spotify
[36] https://www.terraform.io/
[37] https://aws.amazon.com/cloudformation/
[38] https://convox.com/
[39] https://aws.amazon.com/fargate/
[40] https://aws.amazon.com/fargate/
[41] https://aws.amazon.com/sqs/

55	app_engine, gae, appengine, datastore, manage_vms, internally, google_cloud_platform, google_app_engine, compute_engine, django, disclaimer_work, cloud_sql, manage_vm, snapchat, google_compute_engine, container_engine, emulate, deprecate, runtimes, cloud_storage, ga, par, pubsub, private_network, task_queue, rabbitmq, flexible_environment, cloudsql, rel, nofollow	Related to Google Cloud services, such as AppEngine, PubSub and Compute Engine.	Outsourcing operational activities to serverless/cloud providers

Table 4.3: Interpretation of NoOps-related topics and their mapping to the literature findings

Based on the information provided in Table 4.3, it can be concluded that the following literature findings are supported by the practitioners' viewpoint. The numbers in the brackets represent how many of the NoOps-related topics are mapped to that specific finding.

- Technical characteristics
  - Outsourcing operational activities to serverless/cloud providers (10/17)
  - Everything-as-Code (3/17)
  - Auto-scaling (3/17)
  - Orchestration mechanisms (2/17)
  - Automated deployment (1/17)
  - Automated monitoring (1/17)
  - Continuous Integration tooling (1/17)
  - Automated testing (1/17)
  - Automated infrastructure provisioning (1/17)
- Cultural characteristics
  - Removing the silo between Dev and IT Ops teams (1/17)
- Technical challenges
  - Vendor lock-in (1/17)

It can be noticed that most of the topics discuss one or more technical characteristics. This is not surprising, considering the fact that most topics refer to certain cloud-related products and services. It can also be observed that no information related to the NoOps advantages could be inferred, and that only one topic was related to a NoOps-specific challenge. This is also due to the fact that the topic modeling approach did not account for sentiment-related information, hence advantages and challenges can not be easily inferred from the available data. To overcome this limitation of the chosen topic modeling approach, opinion mining needs to be performed.

4.3 Opinion mining

As discussed in the previous section, the use of topic modeling resulted in a large number of topics, most of them being related to technical characteristics. This is due to the fact that the online conversations were related mostly to specific tools, rather than discussing high-level NoOps-related aspects, as was the case in practitioners' literature. Since only one NoOps challenge could be identified and not even one NoOps advantage could be inferred, it means that topic modeling is not suitable as a monomethod: the generated topics do not contain enough information to derive strong claims, such as advantages or challenges. This is problematic, since advantages and challenges are the two categories of findings that were mostly reported by companies. Hence, it would be interesting to understand whether these strong claims are actually supported by practitioners. Therefore, the approach presented in this section aims at addressing the topic modeling limitations, by focusing on inferring NoOps-related advantages and challenges.

One approach that "analyzes people's opinions, appraisals, attitudes, and emotions towards entities, individuals, issues, events, topics, and their attributes" is opinion mining, according to Liu [56]. Opinion mining involves the use of sentiment analysis [53], which commonly refers to the approach of detecting the polarity of a text (i.e., whether it is positive, negative or neutral) [64]. Since our goal is to analyze the attitudes expressed in online discussions regarding certain NoOps-related attributes, opinion mining is a suitable approach.

In software engineering research, opinion mining has been used for a wide variety of applications, ranging from assessing software quality [8], to understanding product reviews [36, 40, 69], and even to analyzing emotional patterns of developers during software development activities [35]. The rest of this section is structured as follows. Subsection 4.3.1 introduces the opinion mining approach, as well as some related studies. Then, subsection 4.3.2 presents the opinion mining results.

4.3.1 The opinion mining approach

The opinion mining approach applied in this study can be described as follows. First, all comments that contain NoOps-related terms are extracted from the dataset, using the NoOps-related topics obtained via topic modeling. Then, the comments undergo a preprocessing procedure, followed by the application of sentiment analysis tools in order to detect the opinionated comments. Afterwards, the opinionated comments are manually mapped to corresponding literature findings, with the purpose of confirming or refuting the literature findings. Each of these steps will be described in detail below.

Extraction of NoOps-related comments

The previously discussed topic modeling approach resulted in 17 different topics that were classified as NoOps-related, meaning that those topics contain words that could be directly mapped to NoOps-specific concepts (as shown in section 4.2.2). This implies that the dataset could also contain comments that are not related at all to NoOps. Hence, since we are interested in mining NoOps-related opinions, we can use the NoOps-related topics to extract from the dataset specifically those comments that contain NoOps-specific words.

The application of the topic modeling approach discussed in section 4.2.1 outputs a topic distribution for each comment in the dataset. In other words, each comment is represented as an array of k elements, where k is the number of topics in the model. Each element on position i in the array ($0 \leq i < k$) indicates the proportion of words in a comment that are related to topic i. Hence, a comment can be considered related to NoOps if the sum of the proportions of the NoOps-related topics for that comment is greater than 0. In other words, the comment is related to NoOps if it contains at least one word belonging to at least one of the topics that were marked as related to NoOps.

However, extracting comments that were marked as NoOps-related using a threshold of 0 could result in many comments that have a very low proportion of words that belong to NoOps-related topics. If a comment has a very low proportion of words that belong to NoOps-related topics, this could mean that NoOps might not the main topic of that comment. This claim is strengthened by the fact that the topics that were interpreted as NoOps-related might contain some words that are not related at all to NoOps: the interpretation of the topics was done only based on the top-30 most prevalent words in a topic, despite the fact that a topic could contain more than 30 words. While some of these less prevalent words (beyond top-30) could still be related to NoOps, it could be the case that some of them are not related to NoOps at all. Hence, if an extracted comment contains a very low proportion of words belonging to NoOps-related topics, and if these words are in fact not related to NoOps, it means that the comment might be falsely flagged as related to NoOps. Therefore, to avoid the extraction of comments where NoOps is not the main discussion topic, a threshold greater than 0 needs to be chosen.

The issue is that choosing a proper threshold without compromising the validity of the results is very tricky. As shown in Figure 4.2, the higher the threshold is, the lower the number of extracted NoOps-related comments becomes. At the same time, it can be observed that when the threshold increases, the average number of words in the extracted NoOps-related comments decreases. Hence, it can be assumed that having shorter comments means that there is a lower volume of information for deriving whether a comment is opinionated, meaning that the quality of the results might decrease. Based on this assumption, it can be inferred from Figure 4.2 that the optimal threshold could be 0.3, since it is the point after which the average comment length drops significantly (average comment length is 60 for threshold 0.3, and 40 for threshold 0.4). However, there is no way to prove that that this assumption is true, so this threshold choice can be considered arbitrary. Therefore, we need to understand whether the choice of the threshold actually has any impact on the conclusions of this study. To do this, the analysis will be conducted on data extracted using three different thresholds, namely 0.2, 0.3 and 0.4.

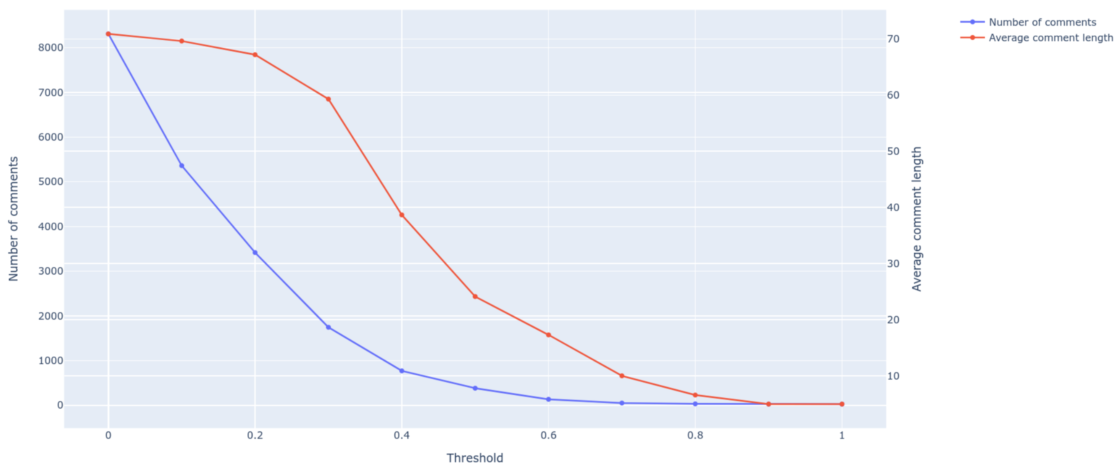


Figure 4.2: The number of NoOps-related comments and the average length of NoOps-related comments (number of words) for different thresholds

Assessing the sentiment polarity of comments

There are many approaches that can be used for opinion mining, two of the most common ones being SentiStrength [86] and VADER [39]. SentiStrength is a tool that can be used to determine the polarity of a short text (whether it is positive, negative or neutral). It is based on a lexicon where each word is augmented with a certain strength. Then, it assigns the strength for each word in a sentence, the final polarity of the sentence representing the sum of all scores. VADER is another popular tool which uses a lexicon and rule-based approach to determine the polarity of social media texts at the sentence level.

In the PhD dissertation of Bin Lin [53], some common issues related to the application of opinion mining tools to software engineering data are discussed. One of the most common issues is related to using opinion mining techniques out-of-the-box. As shown in multiple studies, applying opinion mining tools to software engineering data usually results in inaccurate predictions [44, 87], since most of the tools are not trained on this kind of data. Since the NoOps-related comments analyzed in this study are related to software engineering, it is important to account for this issue as well.

To overcome the aforementioned issue, some tools for performing opinion mining on software engineering data have been proposed. One tool is SentiStrength-SE [41], which is an enhancement of SentiStrength [86] with a lexicon adapted to software engineering texts. Other tools are trained for specific software engineering artifacts, such as code reviews (SentiCR [2]) and StackOverflow data (Senti4SD [14]).

Since our NoOps-related comments originate from online discussions on social news websites, we can not expect the conversations to be full of technical details, as is the case on StackOverflow or in code reviews. Hence, it could be the case that tools such as SentiCR and Senti4SD are not suitable for this kind of data. However, since SentiStrength-SE uses the original lexicon of SentiStrength (which was created based on informal social media conversations on MySpace), but augmented with software engineering data, it means that this tool could be suitable for this study. In addition, the evaluation of VADER shows that it performs better than competitor approaches on short social media text (tweets), as well as product reviews, hence VADER could be a suitable candidate for the current task as well. However, as discussed by Lin et al. [54], there is no tool that is completely ready for determining sentiment polarity in software engineering discussions. Hence, to reduce the risk of errors, we made the decision of using both tools in parallel, and continue the analysis only for the cases where both tools agree.

The use of these tools implies that the input data (comments) needs to satisfy certain requirements. Although SentiStrength-SE does not require the user to perform any preprocessing steps beforehand [41], it seems that papers that used VADER employ preprocessing techniques [23, 70]. More specifically, the preprocessing is conducted as follows. First, all HTML entities are parsed and all hyperlinks are removed. Then, all numbers are removed, as well as mentions and hashtags. The next step involved expanding all contractions (e.g., “won’t” becomes “will not”, “can’t” becomes “can not”, etc). Finally, all comments are split into sentences, as suggested in the VADER documentation, using the NLTK sentence tokenizer.[42]

After the preprocessing step, both SentiStrength-SE and VADER are applied on the preprocessed sentences. The results are interpreted as follows. For SentiStrength-SE, the tool reports two scores for each sentence. One score is related to the positive sentiment, ranging from 1 (not positive) to 5 (extremely positive), while the other is related to the negative sentiment, ranging from -1 (not negative) to -5 (extremely negative) [53]. These two values are summed, and the result is interpreted following the approach of Lin et al. [54]: positive polarity, if the sum is greater than 0, neutral, if the sum is equal to 0, and negative if the sum is below 0. For VADER, four different scores are reported: positive, neutral, negative and compound. All these scores range from 0 to 1, with the exception of the compound score, which is normalized between -1 and 1. The result is interpreted based on the compound score according to the suggestion of VADER’s author,[43] as follows. If the compound score is higher than or equal to 0.5, the sentence is positive. If the compound score is lower than or equal to -0.5, the sentence is negative. However, if the compound score is between 0.5 and -0.5, then the sentence has a neutral polarity.

In order to understand the degree of agreement between the tools and human annotators, we extracted a random sample of 32 sentences from the dataset. Then, each of these sentences was manually annotated based on the subjective interpretation of its polarity, following the example of Fucci et al. [28]. The sample size of 32 sentences has been chosen in accordance to the study of

[42] `http://www.nltk.org/howto/sentiment.html`

[43] The suggestions can be found here: `https://github.com/cjhutto/vaderSentiment`

Fucci et al. [28]. Since this annotation process is a subjective process, the creation of a theoretical framework was needed in order to ensure reproducibility. This theoretical framework was adapted from the study of Fucci et al. [28], and consists of the following rules:

- A sentence is labeled "positive" if it contains at least one positive remark about at least one aspect, and there are no negative remarks about any aspect (e.g., "switching to serverless lowered our costs with operations").
- A sentence is labeled "negative" if it contains at least one negative remark about at least one aspect, and there are no positive remarks about any aspect (e.g., "NoOps is just a vendor lock-in trap").
- A sentence is labeled "neutral" if it does not contain any positive or negative remarks about any aspect (e.g., "we use lambda to automate simple use cases").
- A sentence is labeled "mixed" if it contains both positive and negative remarks about any aspect (e.g., "azure is really easy to learn but I hate the ui").

The manual annotation was conducted solely by the author of this thesis. During the annotation process, the annotator did not look at the results of the tools, in order to avoid introducing potential biases.

Analysis of opinionated comments

After determining the sentence-level polarity of the data and extracting all cases where both tools agreed, the next step was to eliminate all sentences that were labeled as "neutral". In order to identify the comments that could be related to NoOps advantages or challenges, we need to extract all sentences that reflect a non-neutral sentiment polarity, since it is important to understand what are the aspects that practitioners agree or disagree with.

After extracting all sentences with a non-neutral sentiment polarity (positive or negative), the analysis was conducted as follows. Each sentence of non-neutral polarity was manually mapped to specific advantages or challenges identified from the literature study (see chapter 3). Then, for each sentence - finding mapping, it was recorded whether the sentence confirms or refutes the finding. For example, for the literature finding that NoOps leads to cost reductions we identified sentences that confirmed this finding: one reason is that it is much cheaper to outsource operations to cloud vendors, rather than pay the salaries of a dedicated operations team. Other sentences refuted this finding, claiming that the migration towards the cloud actually led to higher costs than before, since some employees made a lot of mistakes which skyrocketed the costs. The sentences that could not be interpreted at all, or those that could not be mapped to any literature finding, were marked as "N/A".

It is important to note that the dataset also contained sentences that could not be interpreted individually, due to the lack of context. For instance, consider the comment "Switching to serverless lowered our costs with operations. That's amazing!". This comment is composed of two sentences. While both sentences are positive, analyzing the second sentence individually does not reveal any information. Hence, whenever these situations were encountered, those sentences were interpreted based on the full content of the parent comment.

4.3.2 Findings

This subsection introduces the results of the analysis of opinionated comments related to NoOps advantages and challenges. These results will be discussed in more detail in section 4.4.

Comment selection and preprocessing

After applying the threshold-based comment selection procedure, the following results were obtained. For threshold 0.2, the total number of extracted comments is 3418, 1764 for threshold

0.3, and 771 for threshold 0.4. Moreover, the sentence tokenization preprocessing step resulted in 10929 sentences for threshold 0.2, 5201 sentences for threshold 0.3 and 1678 sentences for threshold 0.4.

Evaluating the suitability of the sentiment analysis tools

Table 4.4 shows some statistics related to the application of both SentiStrength-SE and VADER on data extracted using three thresholds. It can be observed that the application of both tools resulted in a tool agreement of 74% for threshold 0.2, 76% for threshold 0.3, and 79% for threshold 0.4. To better understand these disagreements, we follow the example of Novielli et al. [65] and distinguish between severe and mild disagreements. Severe disagreements represent the cases when one tool reports "positive", while the other reports "negative". Mild disagreements represent the cases when one tool reports "neutral", while the other reports either "positive" or "negative". The results show that for threshold 0.2 there are 152 severe disagreements (1.3% of the dataset), 58 for threshold 0.3 (1.1% of the dataset), and 9 for threshold 0.4 (0.05% of the dataset). Mild disagreements are the most common ones, with 2591 cases for threshold 0.2 (23.7% of the dataset), 1180 for threshold 0.3 (22.6% of the dataset), and 340 cases for threshold 0.4 (20.2% of the dataset). To interpret these percentages, they can be compared to the results presented in the paper of Novielli et al. [65], where a comparison between multiple sentiment analysis tools is performed on multiple datasets. In this paper, the authors report severe disagreements rates ranging from 2% (for a pull request comments dataset), up to 16% (for pull request discussions and commit discussions datasets). For mild disagreements, the authors report percentages ranging from 14% (for a pull request comments dataset), up to 30% (for a pull request discussions dataset). Although Novielli et al. do not compare the disagreements of SentiStrength-SE and VADER specifically, their results are still useful to understand what kind of performance can be expected from using this kind of tools. The percentages for severe disagreements obtained on our dataset seem to be lower than any of the percentages reported by Novielli et al., while the percentages obtained for mild disagreements seem to be aligned with their findings. Based on these results, it can be inferred that the combination of SentiStrength-SE and VADER could actually be suitable for our data.

	Threshold 0.2		**Threshold 0.3**		**Threshold 0.4**	
	SentiStrength-SE	**VADER**	**SentiStrength-SE**	**VADER**	**SentiStrength-SE**	**VADER**
Positive	1511	1943	709	904	213	256
Neutral	8352	8414	4071	4071	1366	1376
Negative	1065	572	421	226	99	46
Severe disagreements	152		58		9	
Mild disagreements	2591		1180		340	
Inter-tool agreement	0.748		0.761		0.792	

Table 4.4: Statistics related to the application of SentiStrength-SE and VADER on data extracted using three thresholds

Moreover, the evaluation of the degree of agreement between the tools and human annotators resulted in an agreement of 87.5% (there were 4 disagreements, out of 32 labeled sentences). All disagreements, however, were of mild nature. For instance, the sentence "Docker works wonders if you dont have a functioning app server or something equivalent" was marked neutral by both tools, whereas it is positive, since it describes a situation where Docker would be a good fit. However, the fact that no severe disagreements existed shows that the choice of tools could be suitable for this dataset.

Analysis of opinionated comments

Table 4.5 shows the results of the analysis of comments related to NoOps advantages. The table highlights a few main points. First, it seems that the threshold choice has a solid impact on the results - although for threshold 0.2 there are 11 out of 17 advantages that were observed in the extracted comments, for threshold 0.4 there is only one. In other words, the threshold has an impact on the completeness of the results. This implies that one should also consider the usage of lower thresholds (e.g., 0.1 or even 0.05). Throughout the rest of this chapter, the results for a threshold of 0.2 will be discussed. Second, the most commonly observed advantage (19 sentences) is that NoOps provides "cost savings for infrastructure and application operations". This challenge is both supported and refuted by practitioners, with 8 supporting sentences and 11 contradictory. Third, there is only one other benefit that is disputed by practitioners, namely the benefit that NoOps enhances reliability. Fourth, there are 6 benefits that were not observed at all in the analyzed dataset. However, this is not surprising, considering the fact that these advantages also had a low frequency rate within the analyzed gray literature. Hence, their validity is questionable. Moreover, the fact that 11 out of 17 advantages are supported by practitioners shows that practitioners tend to agree with gray literature.

		Threshold 0.2			**Threshold 0.3**			**Threshold 0.4**		
NoOps advantage reported in literature	**P**	**T**	**C**	**R**	**T**	**C**	**R**	**T**	**C**	**R**
Facilitates testing automation	1	0	0	0	0	0	0	0	0	0
Lowers the risk for cyberattacks	1	2	2	0	1	1	0	0	0	0
Enables self-service	1	0	0	0	0	0	0	0	0	0
Increases accountability	1	1	1	0	0	0	0	0	0	0
Build in Governance, Observability, Quality and Compliance	1	0	0	0	0	0	0	0	0	0
Improves collaboration	2	1	1	0	1	1	0	0	0	0
Facilitates quality engineering	2	0	0	0	0	0	0	0	0	0
Enhances operations process with further intelligence, reducing the need for incident responses	3	0	0	0	0	0	0	0	0	0
Enhances reliability	3	3	2	1	2	1	1	0	0	0
Incentivizes developers to write better quality code	2	0	0	0	0	0	0	0	0	0
Provides scalability	4	2	2	0	0	0	0	0	0	0
Reduces necessity for scaling ITOps team size	5	3	3	0	3	3	0	2	2	0
Minimizes the risk of human error	8	1	1	0	0	0	0	0	0	0
More time and opportunities for learning and growth for ITOps team	9	1	1	0	0	0	0	0	0	0
Cost savings for infrastructure and application operations	14	19	8	11	10	5	5	0	0	0
Additional operational efficiency for infrastructure and application operations	17	4	4	0	2	2	0	0	0	0
Eliminates friction with the infrastructure operations (especially in the release phase), such that developers can create value by not having to interoperate with operations teams	22	3	3	0	0	0	0	0	0	0

Table 4.5: Results of the analysis of opinionated comments related to NoOps advantages. For each considered threshold, T represents the total number of sentences that could be mapped to a specific advantage, C represents the number of sentences that confirm (support) an advantage, while R represents the number of sentences that refute an advantage. P represents the frequency of each advantage in the analyzed literature and it is added for comparative purposes.

Table 4.6 shows the results of the analysis of comments related to NoOps challenges. Similar to the results for advantages, it can be seen that the threshold has a solid impact on the results - while there are 5 out of 11 challenges identified in the dataset created using a threshold of 0.2, there were only 2 out of 11 challenges in the dataset created using a threshold of 0.4. The most commonly reported challenge is that outsourcing activities to third-parties could lead to technical issues and limitations, which is confirmed 22 times. It is interesting to note that there is one challenge that is completely refuted, namely the one related to the management of functions when using Function-as-a-Service solutions. However, considering that only one sentence was related to this challenge, it can not be concluded that this challenge is completely invalid. Another commonly reported challenge is vendor lock-in, which was confirmed 5 times and rejected 2 times. Finally, it can be observed that the challenge that adoption may not be suitable or could require re-engineering of legacy systems, which was the most commonly reported challenge in practitioners' literature (especially by consultancy companies), was observed only two times in the analyzed comments. This implies that the previously discussed concern, that some of the analyzed literature was published for promotional purposes, could actually be valid. Hence, it could be the case that consultancy companies reported this NoOps suitability challenge frequently just as a way to promote their re-engineering or cloud migration services. However, we do not have enough evidence to conclude that this is indeed the case, especially since this specific challenge was not reported only by consultancy companies.

		Threshold 0.2			Threshold 0.3			Threshold 0.4		
NoOps challenges reported in literature	**P**	**T**	**C**	**R**	**T**	**C**	**R**	**T**	**C**	**R**
NoOps can not be based on traditional ways of working together	1	0	0	0	0	0	0	0	0	0
Reported hesitation to understanding the business in the initial phases of adoption	2	0	0	0	0	0	0	0	0	0
Reported discomfort in the initial phases of adoption	5	5	5	0	1	1	0	0	0	0
Management of functions when using FaaS solutions	1	1	0	1	1	0	1	0	0	0
Poor NoOps implementation quality could lead to cyber risk at scale	3	0	0	0	0	0	0	0	0	0
Vendor lock-in	4	7	5	2	3	2	1	2	2	0
Outsourcing activities to third-parties could lead to technical issues and limitations	11	22	22	0	11	11	0	2	2	0
Adoption may not be suitable or requires re-engineering of legacy systems	30	2	2	0	2	2	0	0	0	0
NoOps requires a mindset shift for developers in order to ensure that the developed artifact behaves the same in both local and production environments	6	0	0	0	0	0	0	0	0	0
Regulatory compliance, security and other cross-cutting concerns need to be addressed company-wide	3	0	0	0	0	0	0	0	0	0
Adopting NoOps could lead to less involvement from development in operations, which completely contradicts DevOps practices	3	0	0	0	0	0	0	0	0	0

Table 4.6: Results of the analysis of opinionated comments related to NoOps challenges. For each considered threshold, T represents the total number of sentences that could be mapped to a specific challenge, C represents the number of sentences that confirm (support) a challenge, while R represents the number of sentences that refute a challenge. P represents the frequency of each challenge in the analyzed literature and it is added for comparative purposes.

4.4 Discussion

This section provides a discussion of the results that emerged from the analysis of NoOps-related comments, using both topic modeling and opinion mining.

The mapping of topics to findings shows that only 9 out of 29 technical characteristics were discussed by practitioners. What is remarkable is the fact that those 9 findings are among the most commonly reported ones in practitioners' literature (as can be seen in Table D.3), which implies that these technical characteristics might actually be representative for NoOps. However, it is surprising that frequently reported literature findings, such as the automation of operations based on artificial intelligence, self-service portals, as well as self-healing software, were not discussed at all by practitioners. The fact that these three characteristics are reported (in literature) by product development companies deepens the enigma even more, since one might expect that characteristics reported by companies that develop their own software would also be discussed by practitioners. However, the fact that these characteristics could not be identified via topic modeling does not necessarily mean that they are invalid, but rather that there is not enough data to infer them or that the research method is not suitable. The topic modeling approach also resulted in one topic that was related to a cultural characteristic (removing the silo between development and operations teams), and one technical challenge (vendor lock-in). The fact that only one cultural characteristic could be inferred could mean that topic modeling might not be the right method for inferring cultural characteristics. In order to obtain a clear picture of the socio-cultural aspects of NoOps, a more suitable approach would probably be an ethnographic study in the industry, such that the socio-cultural characterization would reflect what is really happening, rather than what people claim it happens (as would be the case with what practitioners discuss on public channels). Moreover, the fact that the vendor lock-in challenge was identified is due to the fact that "vendor lock-in" is a syntagm. However, topic modeling is not suitable for inferring challenges that require deeper interpretation, such as "reported discomfort in the initial phases of adoption".

Since topic modeling proved to be unsuitable for inferring advantages and challenges, opinion mining was performed. When it comes to NoOps advantages, the majority of those reported in literature were also discussed by practitioners (11 out of 17). Regarding NoOps challenges, only 5 out of the 11 challenges reported in literature were identified.

While most of the discussed advantages were confirmed by practitioners, two of these advantages were rather controversial. The most controversial is that NoOps provides cost savings for infrastructure and application operations, which was contradicted in 11 different sentences, and supported in 8 sentences. On the one hand, those that contradict this advantage refer strictly to the use of services provided by cloud vendors. One perspective is that if these services have availability issues, they could disrupt the software that uses them, potentially causing highly significant financial losses. The use of third-party cloud services is also often criticized for high costs, and sometimes even for the lack of transparency in billing. One comment discusses the experience of a large company with the integration of serverless solutions within a legacy system. This company made the decision for cloud migration expecting a reduction of the costs associated with the size of the operations team. However, the migration actually increased the costs by three times, especially due to revenue losses caused by reliability issues and the long times required for automatically provisioning the infrastructure. This comment also refutes the reported advantage that NoOps enhances reliability, since it clearly describes that this migration led to significant reliability issues. Another comment describes the case of a legacy product that integrated certain cloud services, despite not optimized for this kind of integration, emphasizing that the operations expenses for this product doubled after the cloud integration, compared to when the product was operated in-house. On the other hand, some of those that support this advantage claim that the cost savings are due to a general reduction of operational complexity. Others claim that the cost reductions are related to the downsizing of operations teams, which is an argument that was also mentioned in practitioners' literature. Another comment mentions that NoOps solutions can definitely reduce operations costs, but only when they are used for very specific use cases, such as those that do not require a high amount of computational power. However, for large-scale applications, this might not always be feasible. Hence, it can be concluded that this advantage is relative - while for some NoOps solutions are a financial nightmare, for others the adoption of NoOps truly seems to reduce costs.

It is interesting to note that all benefits of social nature that were reported in literature were also completely supported by practitioners. These advantages are: increased accountability, improved collaboration, the elimination of friction between developers and operations, and the fact that the adoption of NoOps provides operations teams with more time for learning and growth. All these benefits are discussed as experienced consequences of NoOps adoption. Among benefits related to operational processes, practitioners discussed lowering the risk of cyberattacks, as well as the minimization of human errors. The risk of cyberattacks seems to be lowered due to the integration of cloud services provided by large vendors, which are supposed to have dedicated teams that are continuously improving the security of these services. The minimization of human errors is due to the reduction of operational complexity, since NoOps implies less choices, less people and less manual processes, so less things that could go wrong. The only confirmed technical benefit is that NoOps provides scalability, for the same reason as the one reported in practitioners' literature, namely that cloud services are designed to perform automated scaling.

When it comes to the discussed NoOps challenges, the most common one is that outsourcing activities to third-parties could lead to technical issues and limitations. This challenge is confirmed within all identified sentences, practitioners describing issues such as lack of desired functionality (causing developers to search for workarounds), frequent reliability issues, lack of documentation for certain services, or issues related to the interaction with customer support. Another discussed challenge is that the adoption of NoOps may not be suitable for certain systems, or re-engineering might be required. This challenge was reported by only two sentences, which is surprising, considering that this was the most commonly reported challenge in practitioners' literature (especially by consultancy companies). This implies that although the challenge might be real, the fact that it was mostly reported by consultancy companies, while very few occurrences were observed in discussions between practitioners, could mean that the consultancy companies highlight this challenge mainly for promotional purposes. Another commonly reported challenge is related to the reported discomfort in the initial phases of adoption. Practitioners claim that the reason behind this discomfort is the fact that some tools have a very steep learning curve.

Some of the discussed challenges were more controversial. For instance, the challenge of vendor lock-in was supported 5 times, and refuted 2 times. While most practitioners acknowledge the existence of this problem, others report that the problem can be easily avoided by using platform-agnostic solutions, such as Terraform for managing infrastructure provisioning. Another challenge that was mentioned only one time is related to the difficulties of managing functions while using Function-as-a-Service solutions. This challenge was refuted by single comment, the reason being that the management of functions can be easily achieved using function orchestration mechanisms.

4.5 Threats to validity

This section introduces a number of threats that could influence the validity of the results presented in this study, based on the validity criteria introduced by Wohlin et al. [91].

Construct validity

Construct validity refers to whether the theoretical constructs in the study are correctly represented and interpreted in the operational phases [91] (i.e., whether the measured variables "correspond to the intended meanings of the theoretical terms" [22]).

A major concern is related to the selection of methods. As shown in section 4.2.2, although topic modeling provided a picture of the NoOps-related topics discussed by practitioners, it could also be observed that despite the fact that most of these discussion were related to specific tools and techniques, it could not be inferred how those tools were discussed. For the scope of this study, understanding which tools were discussed was enough to interpret the topics and map them to various technical characteristics, since to achieve this goal, it was only necessary to understand what was being discussed. However, as previously discussed, topic modeling is not suitable for deriving strong conclusions, as is the case with advantages or challenges. The reason is that deriving this kind of conclusions involves a deeper understanding of a semantic context, whereas topic modeling is strictly based on syntactic features. To overcome this limitation, opinion mining was performed, focusing on inferring NoOps advantages and challenges. However, this combination of methods was still not suitable for inferring any insights regarding cultural characteristics, or even regarding the NoOps definition. In other words, the chosen methods might not be perfectly suitable for measuring the intended constructs.

Another threat is related to the definition of the research question addressed in this chapter (RQ4). Since the question refers to how NoOps is described in online discussions, one could think that the goal of this research question is identical to that of the first research question (RQ1, addressed in chapter 3), which was also related to the description of NoOps. However, the goal of RQ4 is completely different, despite the similar formulation. While RQ1 had a limited scope (inferring NoOps definition, advantages, positioning), the scope of RQ4 is much wider, aiming at obtaining an overview of practitioners' viewpoint towards NoOps, for comparative purposes.

Another concern that could affect construct validity is related to the data collection process, which was not conservative. Considering that the data collection focused primarily on discussion threads, although the analysis was conducted on comments, it could be expected that the final dataset would contain a high number of comments that are not related to NoOps. Based on this, as well as the fact that there was absolutely no manual selection of the comments that contained information related to NoOps, one could think that, in the case of topic modeling, the generated topics would just be a random collection of words, making them uninterpretable. However, this was not the case, since most of the generated topics had a very high semantic coherence (intuitively assessed), and interpretation could be done reasonably well. One example is that of topic 38, which was related to a geopolitical discussion related to climate change and economical issues. Despite the fact that this topic is not related to NoOps at all, it shows that the topic modeling approach was capable to differentiate between the various discussions that took place in the selected threads. Considering that the approach was completely unsupervised, the fact that we have a clear picture of the types of discussions that existed in those comments and we are able to differentiate between the types of discussions, is a proof for the validity of the approach. Of course, the topic modeling approach is not perfect. For example, consider the examples of topic 23 and 25. Both of these topics contained words related to costs of using services provided by cloud computing vendors. Although one would expect these topics to be a single one, the model marked these as two separate topics. The fact that there is no method available for visualizing the inter-topic similarity (similar to LDAVis [82], which is created specifically for LDA), makes the accurate interpretation of the topics even harder, since it becomes impossible to understand the relations between the topics, thus limiting the interpretation only to the most frequent words of a topic.

Internal validity

Internal validity refers to whether the relationships between the treatment and the outcome are justified, and are not the result of a neglected factor [91].

One threat for internal validity is that the data extraction phase during opinion mining was based on a threshold. In order to understand how the threshold influenced the results, we decided to conduct the analysis for three different thresholds. As previously shown, the threshold had a major impact on the results, especially regarding completeness - a high threshold resulted in a small number of findings, while a lower threshold resulted in a higher variety of findings. Although the findings presented in this chapter were based on threshold 0.2, it could be the case that a lower threshold (e.g., 0.1 or 0.05) could result in a wider array of NoOps advantages and challenges, which could potentially strengthen the conclusions. Hence, the fact that this threshold could eliminate insightful data implies that the conclusions derived based on opinion mining might represent the viewpoint of only a subset of the practitioners that engaged in NoOps-related discussions on the selected social news websites. Consequently, external validity could also be affected, since the conclusions might not be generalizable for the entire population of practitioners.

Another threat is that the collected comments contained many language errors, which might have affected the results of both topic modeling and opinion mining. Although the use of an automatic spelling corrector was attempted on a small sample, the results were highly inaccurate, which led to the decision of leaving the data unaltered. Moreover, some comments also contained sentence formatting errors (e.g., no whitespace after some punctuation marks), which might have affected the output of the sentence tokenization step that was performed during opinion mining. Furthermore, some of the analyzed comments contained references to certain cloud services, for which naming was not consistent. For instance, looking at topics 9 and 13 from the topic modeling results, we see that both topics contain references to AWS Elastic Beanstalk. However, while topic 9 refers to the service as either "elastic_beanstalk" or just "beanstalk", topic 13 refers to it as "eb", which means that practitioners might refer to the same service in different ways. Hence, this issue might have a negative impact on the topic modeling results.

Subjectivity is also a major threat. Since manual interpretation was conducted at multiple stages by a single researcher (e.g., relevance-based filtering during data collection, interpretation of topics, manual validation of sentiment analysis tools, as well as the manual analysis of comments), there is a risk that reasoning errors were introduced, potentially affecting the validity of the results. To reduce subjectivity, theoretical frameworks were created for some of these subjective stages, although these frameworks were not validated objectively.

External validity

External validity refers to whether the results of this study can be generalized to a wider context [91].

One concern is related to the data collection process. Although the search for discussions was performed on two popular social news websites, this does not mean that these are the only two platforms where practitioners might discuss about NoOps. Hence, the generalizability of the results could be affected, since the results might be representative only for a small segment of practitioners.

Chapter 5

Conclusions

In this thesis, a mixed methods study was conducted in order to obtain an exploratory overview of the NoOps concept. This study is motivated by the fact that there are no academic studies regarding NoOps, despite the increasing popularity of the concept in the online media. The main goal of this thesis is to identify the different dimensions of the NoOps definition, the positioning of the concept, the associated characteristics, as well as the reported advantages and challenges. To achieve this goal, we accounted for the perspective of the various contributors to the public discourse, analyzing both practitioners' literature and online discussions. Moreover, this study aims at bridging the gap between industrial practice and academic research.

The first step in reaching the main goal of this thesis involved conducting a systematic literature mapping study, which focused primarily on practitioners' literature. We followed the systematic mapping approach proposed by Petersen et al. [71], as well as the guidelines of Garousi et al. [30] for including practitioners' literature in systematic literature studies. Based on this approach, we identified 98 primary studies, which were subject to manual qualitative analysis. The results show that NoOps is gaining popularity in practitioners' literature. Moreover, the analysis enables us to identify multiple components of the NoOps definition, the positioning of the concept, multiple types of technical and cultural characteristics, as well as reported advantages and challenges.

Regarding the definition, we identified 5 different components, based on which a general definition for NoOps was proposed. In terms of positioning, we showed that NoOps is unanimously described as being an augmentation of DevOps. Furthermore, multiple categories of NoOps advantages were identified, these categories being related to operational processes, organization and business, but also to social and technical aspects. Additionally, multiple types of technical characteristics were inferred, these being related to automation aspects, tooling, architectural features, but also practices. Cultural characteristics were also inferred, the categories being related to collaborative and organizational aspects, mindset changes, and shifting of responsibility. Finally, multiple types of challenges were also distilled, including social, technical, cultural, organizational and general challenges.

Additionally, we found out that some findings tend to be reflected more by certain kinds of organizations, which triggered some questions regarding the validity of some findings. To address this validity issue, we decided to triangulate the findings by obtaining an overview of practitioners' viewpoint about NoOps, as expressed in online discussions on social news websites. To obtain an overview of what is being discussed, we first applied topic modeling. The topic modeling results highlighted 55 different topics, out of which 17 were marked as related to NoOps. These 17 topics were mapped to the existing literature findings. This mapping allowed us to confirm 9 different technical characteristics, one cultural characteristic, and one technical challenge. Since topic modeling alone proved to be insufficient for inferring how certain aspects are discussed, we performed opinion mining, focusing on inferring NoOps advantages and challenges. The results show that most of the advantages reported in literature are also discussed by practitioners, the vast majority of advantages being unanimously supported. Several controversial advantages have also been identified, namely the advantages that NoOps adoption results in cost savings, but also that NoOps enhances reliability. Based on the claims of practitioners, these advantages are rather contextual. Furthermore, the results related to challenges show that practitioners discussed mostly technical and social challenges. In terms of technical challenges, practitioners discussed 3 out of 4 challenges reported in literature, two of them being supported, and one being refuted. Regarding social challenges, practitioners unanimously support the finding that NoOps adoption causes discomfort, especially in the initial adoption phases. Practitioners unanimously agree with the technical challenges that outsourcing operational activities to third-parties could lead to technical issues and limitations, but also that adoption may not be suitable for any kind of system, re-engineering possibly being required. Moreover, the challenge of vendor lock-in is disputed - while some believe that it is a real issue, others believe that it can be avoided. Finally, the fact that only a small number of findings could be validated does not mean that we can conclude that those findings are completely invalid; this rather shows that the chosen methods and the dataset might have not been the most suitable.

Since this study can be considered exploratory, future work is required in order to address the limitations of this work, as well as to further establish the area of NoOps.

5.1 Future work

Considering that this thesis is the first inquiry into the area of NoOps, there are many directions for future work.

First, additional work is needed in order to address some of the limitations and threats to the validity of this study. To this end, some possible directions could be:

- **Addressing study-wide subjectivity concerns**. Since the entire study was conducted by a single person, significant biases could have been introduced, which could have affected the validity of the results. To address these issues, inter-rater reliability assessment needs to be conducted with multiple raters, for all the subjective steps performed in this study. Consequently, the data should be re-analyzed, in order to account for the introduced changes.

- **Differentiating between expectations and experiences**. One important limitation of the literature study, in particular, is that we did not record which findings are supported by experience or represent pure expectations. Hence, in order to strengthen the conclusions of the study, one possible next step would be to annotate all sources with the types of evidence provided for the claims, to facilitate the differentiation between expectations and experiences.

- **Improving the primary source collection processes**. For the literature study, a possible improvement regarding the primary source collection process would be to include multiple search engines for collecting practitioners' literature. For the analysis of online discussions, other websites where practitioners engage in conversations can be considered (e.g., forums), to improve the generalizability of the findings.

- **Accounting for newly published sources**. The data collection process should be repeated for both the literature study and the online discussion study, in order to account for sources published after the initial data collection was completed.

- **Manual qualitative coding of online discussions**. As previously discussed, topic modeling had a number of limitations, which constrained the quality of the findings. To address this issue, manual qualitative coding can be performed instead, in order to account for semantic considerations as well.

- **Repeating the opinion mining procedure for multiple thresholds**. In retrospective, the introduction of a threshold in the analysis of opinionated comments was not a wise decision, as results have shown. Therefore, to account for this limitation, we propose repeating the opinion mining procedure for lower thresholds, in order to improve the completeness of the results, and derive stronger conclusions.

Second, beyond addressing the study limitations, some potential ideas for future work include:

- **Empirical investigations regarding NoOps advantages, cultural characteristics and challenges**. Easterbrook et al. [22] discuss the issue that what people say could be different than what they actually do. Hence, when it comes to cultural aspects, it could be that the representation of culture as described by people is different than what happens in reality. This could also be the case for advantages and challenges. The results of our study on NoOps-related online discussions show that some advantages and challenges are contextual. Therefore, in order to obtain an accurate representation of the socio-cultural aspects, as well as of the advantages and challenges/barriers related to NoOps, industrial ethnographies or case studies should be conducted.
- **Transitioning from DevOps to NoOps**. Since NoOps is positioned unanimously as an augmentation of DevOps, future studies could investigate what are the conditions and steps that organizations need to face in order to perform the transition from DevOps to NoOps. Based on this, a NoOps adoption model could be defined.
- **Empirical studies of NoOps-specific practices**. Another approach would be to empirically investigate NoOps-specific practices (e.g., automated monitoring), in order to explore their potential pains, gains, and their various usage and integration models.

Bibliography

[1] Principles of chaos engineering. https://principlesofchaos.org/. 26

[2] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. Senticr: A customized sentiment analysis tool for code review interactions. In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*, ASE 2017, page 106–111. IEEE Press, 2017. 45

[3] Vander Alves, Nan Niu, Carina Alves, and George Valença. Requirements engineering for software product lines: A systematic literature review. *Information and Software Technology*, 52(8):806 – 820, 2010. 9

[4] Maurício Aniche, Christoph Treude, Igor Steinmacher, Igor Wiese, Gustavo Pinto, Margaret-Anne Storey, and Marco Aurélio Gerosa. How modern news aggregators help development communities shape and share knowledge. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 499–510, 2018. 31

[5] Rob Atenstaedt. Word cloud analysis of the bjgp. *British Journal of General Practice*, 62(596):148–148, 2012. 36

[6] Atlassian. DevOps Culture. https://www.atlassian.com/devops/what-is-devops/devops-culture. 1

[7] Atlassian. What is DevOps? https://www.atlassian.com/devops. 1

[8] Issa Atoum. A novel framework for measuring software quality-in-use based on semantic similarity and sentiment analysis of software reviews. *Journal of King Saud University - Computer and Information Sciences*, 32(1):113–125, 2020. 43

[9] Titus Barik, Brittany Johnson, and Emerson Murphy-Hill. I heart hacker news: Expanding qualitative research findings by analyzing social news websites. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2015, page 882–885, New York, NY, USA, 2015. Association for Computing Machinery. 31

[10] C. Bird. Interviews. In Tim Menzies, Laurie Williams, and Thomas Zimmermann, editors, *Perspectives on Data Science for Software Engineering*, pages 125 – 131. Morgan Kaufmann, Boston, 2016. 9, 29

[11] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012. 34

[12] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 35

[13] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8):1157–1166, 1997. Papers from the Sixth International World Wide Web Conference. 36

[14] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. Sentiment polarity detection for software development. *Empirical Software Engineering*, 23(3):1352–1382, Sep 2017. 45

[15] Giuseppe Cascavilla, Damian A. Tamburri, and Willem-Jan Van Den Heuvel. Cybercrime threat intelligence: A systematic multi-vocal literature review. *Computers & Security*, 105:102258, 2021. 5, 35, 36

[16] Chia-Mei Chen, Dan-Wei Marian Wen, Jun-Jie Fang, Gu-Hsin Lai, and Yi-Hung Liu. A study on security trend based on news analysis. In *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, pages 1–4, 2019. 35

[17] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. 23

[18] Breno B. Nicolau de França, Helvio Jeronimo, and Guilherme Horta Travassos. Characterizing devops by hearing multiple voices. In *Proceedings of the 30th Brazilian Symposium on Software Engineering*, SBES '16, page 53–62, New York, NY, USA, 2016. Association for Computing Machinery. 2

[19] Daniel R Denison. *Corporate culture and organizational effectiveness.* John Wiley & Sons, 1990. 20, 22

[20] Digital.ai. "14th Annual State of Agile Report", May 2020. https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report. 1

[21] Tore Dybå and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9):833–859, 2008. 1

[22] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In Forrest Shull, Janice Singer, and Dag I. K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer London, London, 2008. 28, 30, 31, 51, 55

[23] Shihab Elbagir and Jing Yang. Sentiment analysis on twitter with python's natural language toolkit and vader sentiment analyzer. In *IAENG Transactions on Engineering Sciences*. WORLD SCIENTIFIC, Jan 2020. 45

[24] Luis Escobar, Andreas Grabner, and Rob Jahn. Everything as code, Jan 2021. https://www.dynatrace.com/news/blog/everything-as-code/. 26

[25] Institute for Work & Health. "What researchers mean by... Grey literature.", April 2009. https://www.iwh.on.ca/what-researchers-mean-by/grey-literature. 6

[26] Kevin Forsberg and Harold Mooz. The relationship of system engineering to the project cycle. In *Proceedings of the National Council for Systems Engineering First Annual Conference*, pages 57–61, 1991. 1

[27] Emily Freeman. DevOps, 2019. https://aws.amazon.com/devops/what-is-devops/. 1

[28] G. Fucci, N. Cassee, F. Zampetti, N. Novielli, A. Serebrenik, and M. Di Penta. Waiting around or job half-done? sentiment in self-admitted technical debt. In *2021 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR) (MSR)*, pages 403–414, Los Alamitos, CA, USA, may 2021. IEEE Computer Society. 45, 46

[29] Vahid Garousi, Michael Felderer, and Mika V. Mäntylä. The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, EASE '16, New York, NY, USA, 2016. Association for Computing Machinery. 4

[30] Vahid Garousi, Michael Felderer, and Mika V. Mäntylä. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, 106:101 – 121, 2019. 2, 4, 6, 7, 8, 9, 14, 53

[31] Vahid Garousi and Mika V. Mäntylä. When and what to automate in software testing? a multi-vocal literature review. *Information and Software Technology*, 76:92–117, 2016. 28

[32] Manas Gaur, Ugur Kursuncu, Amanuel Alambo, Amit Sheth, Raminta Daniulaityte, Krishnaprasad Thirunarayan, and Jyotishman Pathak. "let me tell you about your mental health!": Contextualized classification of reddit posts to dsm-5 for web-based intervention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 753–762, New York, NY, USA, 2018. Association for Computing Machinery. 35

[33] Martin Gerlach, Tiago P. Peixoto, and Eduardo G. Altmann. A network approach to topic models. *Science Advances*, 4(7), 2018. 35, 37

[34] Mike Gualtieri, Sandy Carielli, and Cheryl McKinnon. I Don't Want DevOps. I Want NoOps., Nov 2011. `https://go.forrester.com/blogs/11-02-07-i_dont_want_devops_i_want_noops/`. 1

[35] Emitza Guzman and Bernd Bruegge. Towards emotional awareness in software development teams. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2013, page 671–674, New York, NY, USA, 2013. Association for Computing Machinery. 43

[36] Emitza Guzman and Walid Maalej. How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd international requirements engineering conference (RE)*, pages 153–162. IEEE, 2014. 36, 43

[37] Ahmad Fathan Hidayatullah and Muhammad Rifqi Ma'arif. Road traffic topic modeling on twitter using latent dirichlet allocation. In *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 47–52, 2017. 36

[38] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, page 80–88, New York, NY, USA, 2010. Association for Computing Machinery. 35

[39] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014. 44

[40] Claudia Iacob and Rachel Harrison. Retrieving and analyzing mobile apps feature requests from online reviews. In *2013 10th Working Conference on Mining Software Repositories (MSR)*, pages 41–44, 2013. 43

[41] Md Rakibul Islam and Minhaz F. Zibran. Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text. *Journal of Systems and Software*, 145:125–146, 2018. 45

[42] Ramtin Jabbari, Nauman bin Ali, Kai Petersen, and Binish Tanveer. What is devops? a systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, XP '16 Workshops, New York, NY, USA, 2016. Association for Computing Machinery. 1, 3, 25

[43] Byeongki Jeong, Janghyeok Yoon, and Jae-Min Lee. Social media mining for product planning: A product opportunity mining approach based on topic modeling and sentiment analysis. *International Journal of Information Management*, 48:280–290, 2019. 35

[44] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 22(5):2543–2584, Jan 2017. 45

[45] Ambily K K. *DevOps Basics and Variations*, pages 1–11. Apress, Berkeley, CA, 2020. 25

[46] Fernando Kamei, Igor Wiese, Crescencio Lima, Ivanilton Polato, Vilmar Nepomuceno, Waldemar Ferreira, Márcio Ribeiro, Carolline Pena, Bruno Cartaxo, Gustavo Pinto, and Sérgio Soares. Grey literature in software engineering: A critical review. *Information and Software Technology*, page 106609, 2021. 4, 13

[47] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15, 2009. Special Section - Most Cited Articles in 2002 and Regular Research Papers. 4

[48] Barbara Ann Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 07 2007. 4, 7

[49] Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications, 2004. 9, 10

[50] Indika Kumara, Martín Garriga, Angel Urbano Romeu, Dario Di Nucci, Fabio Palomba, Damian Andrew Tamburri, and Willem-Jan van den Heuvel. The do's and don'ts of infrastructure code: A systematic gray literature review. *Information and Software Technology*, 137:106593, 2021. 5, 35

[51] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977. 23

[52] Michael Lawrence. What is proactive monitoring?, Nov 2017. `https://smallbusiness.chron.com/proactive-monitoring-73438.html`. 25, 26

[53] Bin Lin. *Opinion mining for software development*. PhD thesis, Università della Svizzera italiana, 2020. 43, 45

[54] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. Sentiment analysis for software engineering: How far can we go? In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 94–104, 2018. 45

[55] Nathan C. Lindstedt. Structural topic modeling for social scientists: A brief case study with social movement studies literature, 2005–2017. *Social Currents*, 6(4):307–318, 2019. 35

[56] Bing Liu. *Web Data Mining*. Springer Berlin Heidelberg, 2011. 43

[57] Yang Liu and Zhijun Yin. Understanding weight loss via online discussions: Content analysis of reddit posts using topic modeling and word clustering techniques. *J Med Internet Res*, 22(6), Jun 2020. 35

[58] Iraj Lohrasbinasab, Prameet Bhakta Acharya, and Ricardo Colomo-Palacios. Bizdevops: A multivocal literature review. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca, editors, *Computational Science and Its Applications – ICCSA 2020*, pages 698–713, Cham, 2020. Springer International Publishing. 2, 3

[59] Chien Lu, Jaakko Peltonen, and Timo Nummenmaa. Game postmortems vs. developer reddit amas: computational analysis of developer communication. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–7, 2019. 35

[60] Kathleen M. MacQueen, Eleanor McLellan, Kelly Kay, and Bobby Milstein. Codebook development for team-based qualitative analysis. *CAM Journal*, 10(2):31–36, 1998. 10

[61] Michael Mazyar. DevOps: The Ultimate Way to Break Down Silos, December 2018. https://devops.com/devops-the-ultimate-way-to-break-down-silos/. 1

[62] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995. 36

[63] Håvard Myrbakken and Ricardo Colomo-Palacios. Devsecops: A multivocal literature review. In Antonia Mas, Antoni Mesquida, Rory V. O'Connor, Terry Rout, and Alec Dorling, editors, *Software Process Improvement and Capability Determination*, pages 17–29, Cham, 2017. Springer International Publishing. 2, 3

[64] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture*, K-CAP '03, page 70–77, New York, NY, USA, 2003. Association for Computing Machinery. 43

[65] Nicole Novielli, Fabio Calefato, Filippo Lanubile, and Alexander Serebrenik. Assessment of off-the-shelf se-specific sentiment analysis tools: An extended replication study. *Empirical Software Engineering*, 26(4), Jun 2021. 47

[66] Columbia University Mailman School of Public Health. "Content Analysis", 2019. https://www.publichealth.columbia.edu/research/population-health-methods/content-analysis. 10

[67] Justice Opara-Martins, Reza Sahandi, and Feng Tian. Critical review of vendor lock-in and its impact on adoption of cloud computing. In *International Conference on Information Society (i-Society 2014)*, pages 92–97. IEEE, 2014. 27

[68] Annibale Panichella. A systematic comparison of search algorithms for topic modelling—a study on duplicate bug report identification. In Shiva Nejati and Gregory Gay, editors, *Search-Based Software Engineering*, pages 11–26, Cham, 2019. Springer International Publishing. 35, 36

[69] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A Visaggio, Gerardo Canfora, and Harald C Gall. How can i improve my app? classifying user reviews for software maintenance and evolution. In *2015 IEEE international conference on software maintenance and evolution (ICSME)*, pages 281–290. IEEE, 2015. 43

[70] Toni Pano and Rasha Kashef. A complete vader-based sentiment analysis of bitcoin (btc) tweets during the era of covid-19. *Big Data and Cognitive Computing*, 4(4), 2020. 45

[71] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, June 2008. 3, 4, 6, 53

[72] Kai Petersen, Claes Wohlin, and Dejan Baca. The waterfall model in large-scale development. In Frank Bomarius, Markku Oivo, Päivi Jaring, and Pekka Abrahamsson, editors, *Product-Focused Software Process Improvement*, pages 386–400, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. 1

[73] Shari Lawrence Pfleeger and Winifred Menezes. Marketing technology to software practitioners. *IEEE Software*, 17(1):27–33, 2000. 18, 19

[74] Rambabu Posa. Drawbacks of monolith architecture. https://www.oreilly.com/library/view/scala-reactive-programming/9781787288645/c2dd1f56-3407-4bab-942c-d6f2d780bb5e.xhtml. 27

[75] Angel Rivera. DevOps didn't exist when I started as a developer: How this one principle changed my career, Sep 2019. https://circleci.com/blog/devops-did-not-exist/. 1

[76] Winston W. Royce. Managing the development of large software systems. In *Technical Papers of Western Electronic Show and Convention*, WesCon, pages 1–9, August 1970. 1

[77] Huub J.M. Ruël, Tanya Bondarouk, and Stefan Smink. The Waterfall Approach and Requirement Uncertainty: An In-Depth Case Study of an Enterprise Systems Implementation at a Major Airline Company. *International Journal of Information Technology Project Management (IJITPM)*, 1(2):43–60, April 2010. 1

[78] Goutam Kumar Saha. Self-healing software. *Ubiquity*, 2007(March), March 2007. 26

[79] Johnny Saldana. *"The coding manual for qualitative researchers"*. SAGE Publications, London, 2015. 9, 10

[80] Mary Sánchez-Gordón and Ricardo Colomo-Palacios. Characterizing devops culture: A systematic literature review. In Ioannis Stamelos, Rory V. O'Connor, Terry Rout, and Alec Dorling, editors, *Software Process Improvement and Capability Determination*, pages 3–15, Cham, 2018. Springer International Publishing. 3

[81] Mohammad Shahrad, Jonathan Balkind, and David Wentzlaff. Architectural implications of function-as-a-service computing. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 1063–1075, 2019. 27

[82] Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. 51

[83] Jacopo Soldani, Damian Andrew Tamburri, and Willem-Jan Van Den Heuvel. The pains and gains of microservices: A systematic grey literature review. *Journal of Systems and Software*, 146:215–232, 2018. 4, 5, 6

[84] VB Staff. Lightspin: 46% of aws s3 buckets could be misconfigured and unsafe, Jun 2021. https://venturebeat.com/2021/06/14/lightspin-46-of-aws-s3-buckets-could-be-misconfigured-and-unsafe/. 27

[85] Bjørnar Tessem and Jon Iden. Cooperation between developers and operations in software engineering projects. In *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pages 105–108, 2008. 1, 17

[86] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010. 44, 45

[87] Parastou Tourani, Yujuan Jiang, and B. Adams. Monitoring sentiment in open source mailing lists: exploratory study on the apache ecosystem. In *CASCON*, 2014. 45

[88] Ike Vayansky and Sathish A.P. Kumar. A review of topic modeling methods. *Information Systems*, 94:101582, 2020. 35

[89] Wikipedia contributors. Social news website — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Social_news_website&oldid=1024399871, 2021. 31

[90] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, New York, NY, USA, 2014. Association for Computing Machinery. 4

[91] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012. 28, 29, 30, 51, 52

[92] A. Yasin, R. Fatima, L. Wen, W. Afzal, M. Azhar, and R. Torkar. On using grey literature and google scholar in systematic literature reviews in software engineering. *IEEE Access*, 8:36226–36243, 2020. 7

[93] T. Zimmermann. Card-sorting: From text to themes. In Tim Menzies, Laurie Williams, and Thomas Zimmermann, editors, *Perspectives on Data Science for Software Engineering*, pages 137 – 141. Morgan Kaufmann, Boston, 2016. 11

List of primary sources

[R1] Anita Engleder. How to get from devops to noops: 5 steps — the enterprisers project, Mar 2020. https://enterprisersproject.com/article/2020/3/how-get-devops-noops-5-steps. 27

[R2] Jason Grosz. From noops to allops - test double — our blog, Nov 2019. https://blog.testdouble.com/posts/2019-11-26-from-noops-to-allops/. 26, 27

[R3] Christopher Hambridge. Moving from devops to noops with a microservice architecture on ..., Jun 2016. https://www.ibm.com/blogs/cloud-archive/2016/06/moving-devops-noops-microservice-architecture-bluemix/. 26

[R4] Mindtree. The evolution of noops- whitepaper on driving more value with ..., Oct 2020. https://www.mindtree.com/about/resources/evolution-of-noops. 25

[R5] Rik Marselis. Quality for devops teams, Mar 2020. https://books.google.nl/books?id=ICHXDwAAQBAJ&pg=PT36&lpg=PT36&dq=%22NoOps%22&source=bl&ots=-0ppcvfKMJ&sig=ACfU3U32waKW1NX8lLgKkH5gMixz46n5rQ&hl=en&sa=X&ved=2ahUKEwiSwLvbmr3tAhXnz4UKHb7sBME4lgEQ6AEwCXoECAAQAQ.

[R6] Venky Chennapragada. No-ops: How a devops strategy centered on automation can ..., May 2020. https://www.capgemini.com/2020/05/no-ops-how-a-devops-strategy-centered-on-automation-can-guarantee-continuous-operations-in 25, 26

[R7] Andreas Grabner. Autonomous cloud enablement aka scaling noops via self-service ..., Feb 2020. https://www.dynatrace.com/news/blog/autonomous-cloud-enablement-aka-scaling-noops-via-self-service/. 25

[R8] Kieran Norton Ken Corless, Mike Kavis. Noops, devsecops, and managing cyber risk - . . . , Feb 2019. https://deloitte.wsj.com/riskandcompliance/2019/02/27/noops-devsecops-and-managing-cyber-risk/. 25

[R9] Bernd Greifeneder. Digital transformation: How we took noops to the . . . , Feb 2020. https://enterprisersproject.com/article/2020/2/digital-transformation-how-noops-helps. 26, 27, 28

[R10] Adam Eivy. wary of the economics of" serverless" cloud computing, Apr 2017. https://ieeexplore.ieee.org/abstract/document/7912239/. 26

[R11] Mike Ensor. Introducing a no-ops culture to remove bottlenecks from cloud-native ..., Jun 2019. https://www.youtube.com/watch?v=eiHgx-pyg1U. 25, 26, 28

[R12] Jordan Bach. Is noops the end of devops? think again — blog — appdynamics, Apr 2017. https://www.appdynamics.com/blog/engineering/is-noops-the-end-of-devops-think-again/.

[R13] Kieran Norton Ken Corless, Mike Kavis. Noops in a serverless world — deloitte insights, Jan 2019. https://www2.deloitte.com/us/en/insights/focus/tech-trends/2019/noops-serverless-computing-transforming-it-operations.html. 21, 25, 26, 27

[R14] Jayashree Hegde. What is noops? is it agile ops? - botmetric, May 2017. https://www.botmetric.com/blog/what-is-noops-agile-ops/. 25, 26

[R15] Harbinder Kang. Devops ventured, noops gained: The natural progression of ..., May 2019. https://www.fusionfabric.cloud/resources/news-insights/devops-ventured-noops-gained-natural-progression-automation-through. 26

[R16] Rotem Dafni. Noops - is this the end of devops as we know it? - - stratoscale, Oct 2016. https://www.stratoscale.com/blog/devops/noops-end-devops-know/. 25, 26, 27

[R17] Chief Information Officers Council. [pdf] white paper: demystifying noops and serverless computing - cio.gov, Jan 2020. https://www.cio.gov/resources/Demystifying%20NoOps%20and%20Serverless%20Computing_FINAL.pdf. 27

[R18] Jason Bloomberg. Newops: The future of operations in a hybrid it world — avnetwork, Mar 2019. https://www.avnetwork.com/blogs/newops-the-future-of-operations-in-a-hybrid-it-world.

[R19] Smartface. What is noops and how to switch from devops to noops in mobile ..., Apr 2018. https://smartface.io/what-is-noops-devops-mobile-app-dev/.

[R20] Adam Serediuk. How to create a successful noops team — xmatters, Nov 2015. https://www.xmatters.com/blog/devops/how-to-create-a-successful-noops-team/. 26

[R21] David Linthicum. Beware the siren song of no-ops — infoworld, Feb 2019. https://www.infoworld.com/article/3341326/beware-the-siren-song-of-no-ops.html.

[R22] Jayashree Hegde. The perspective: Noops through the eyes of a devops engineer, Jan 2017. https://www.botmetric.com/blog/noops-through-eyes-devops-engineer/.

[R23] Kenopsy Services. Noops – the end of devops? – kenopsy, Apr 2020. https://kenopsy.com/noops-the-end-of-devops/. 26

[R24] Tangui Colin. Software factory #2: What makes infrastructure crucial - witekio ..., Sep 2017. https://witekio.com/software-factory-2-what-makes-infrastructure-crucial/. 28

[R25] B.S. Farroha. framework for managing mission needs, compliance, and trust in the devops environment, Oct 2014. https://ieeexplore.ieee.org/abstract/document/6956773/.

[R26] Bernd Greifeneder. Why culture is more important than tech adoption in a software-driven world, Jun 2019. https://enterprisersproject.com/article/2019/6/why-culture-more-important-tech-adoption-software-driven-world.

[R27] Hussein Badakhchani. Redefining noops to better inform it decision-making, Aug 2017. https://devops.com/redefining-noops-better-inform-decision-making/. 26

[R28] Laurent Bride. Serverless computing: Moving from devops to noops, Jun 2018. https://devops.com/serverless-computing-moving-from-devops-to-noops/. 25

[R29] Christian Beedgen. Logging and noops with christian beedgen - software engineering ..., Apr 2016. https://softwareengineeringdaily.com/2016/04/11/logging-noops-christian-beedgen/. 25, 27

[R30] Sysco Labs. Tech debate: Devops vs no ops - sysco labs, Mar 2018. https://syscolabs.lk/blog/tech-debate-devops-vs-no-ops/. 25, 26

[R31] Bob Rhubart. Podcast: Devops to noops: State of play — oracle ..., Sep 2018. https://blogs.oracle.com/developers/podcast.

[R32] Mindtree. Cloud thought leadership series – part 4: Going . . . , Dec 2020. https://www.mindtree.com/about/resources/cloud-native-transformation.

[R33] Mary K. Pratt. What is noops? the quest for fully automated it operations — cio, Jul 2017. https://www.cio.com/article/3407714/what-is-noops-the-quest-for-fully-automated-it-operations.html. 26

[R34] Engine Yard. How noops improves development productivity and operational ..., Sep 2020. https://blog.engineyard.com/how-noops-improves-development-productivity-and-operational-agility. 26

[R35] Scott Brindamour. Is noops achievable? - lumen, Jul 2020. https://blog.lumen.com/is-noops-achievable/.

[R36] Christina Cardoza. A noops state of mind - sd times, Mar 2019. https://sdtimes.com/devops/a-noops-state-of-mind/.

[R37] Keith Townsend. Noops: How serverless architecture introduces a third mode of it ..., Jan 2018. https://www.techrepublic.com/article/noops-how-serverless-architecture-introduces-a-third-mode-of-it-operations/. 27

[R38] E-Spincorp. Noops: Importance and its benefits in it operations — e-spin group, Mar 2019. https://www.e-spincorp.com/noops-importance-and-its-benefits-in-it-operations/. 26

[R39] Michael Allen. Look mum, noops! how to empower the next evolution of it ..., May 2020. https://www.itproportal.com/features/look-mum-noops-how-to-empower-the-next-evolution-of-it-operations/. 28

[R40] Ruben van der Zwan. From devops to noops – how to automate your chocolate factory, May 2017. https://www.yenlo.com/blog/from-devops-to-noops-how-to-automate-your-chocolate-factory.

[R41] Ofir Nachmani. Your cloud management team: Ops, devops or noops ? - iod, Dec 2011. https://iamondemand.com/blog/your-cloud-management-team-ops-devops-or-noops/.

[R42] Adrian Cockcroft. Ops, devops and paas (noops) at netflix - adrian cockcroft's blog, Mar 2012. http://perfcap.blogspot.com/2012/03/ops-devops-and-noops-at-netflix.html. 26

[R43] Mark Boyd. The road to noops: Serverless computing is quickly gaining ..., May 2016. https://thenewstack.io/serverless-computing-growing-quickly/.

[R44] Mia-Platform Team. Devops, dataops, gitops, noops ... let's clarify! - mia platform, Feb 2020. https://blog.mia-platform.eu/en/devops-dataops-gitops-noops-...-lets-clarify.

[R45] Adron Hall. Noops, appops, devops, & more - removing the os barrier with ..., Feb 2012. https://blog.newrelic.com/engineering/noops-appops-devops-more-removing-the-os-barrier-with-paas-part-3/.

[R46] Siva Perubotla. Noops – reality or just another fad — brillio technologies, Mar 2019. https://www.brillio.com/insights/noops-reality-or-just-another-fad/.

[R47] Don Macvittie. No-ops: It's been 5 years - devops.com, Dec 2017. https://devops.com/no-ops-5-years/.

[R48] Forrest Brazeal. Serverless superheroes: Adam johnson, monitoring, and the "no ..., Sep 2017. https://www.google.com/url?q=https://read.acloud.guru/serverless-superheroes-adam-johnson-monitoring-and-the-no-ops-myth-117e468416a2&sa=U&ved=2ahUKEwiO5be1n73tAhVMzIUKHVcJDtU4ggEQFjAGegQIBRAB&usg=AOvVaw3VHfV-T5mVCHnR7hvhXAle.

[R49] Tal Bereznitskey. The blueprints of a "no-ops" startup — hacker noon, Aug 2017. https://hackernoon.com/the-blueprints-of-a-no-ops-startup-e3cf0c9229d2.

[R50] Ellen Messmer. Netflix uses lots of cloud services – but don't call it 'noops ..., Mar 2012. https://www.computerworld.com/article/2502230/netflix-uses-lots-of-cloud-services----but-don-t-call-it--noops-.html.

[R51] Colin But. Noops is a no-no. in my previous post about the evolution... — by ..., May 2018. https://medium.com/@Colin_But/noops-is-a-no-no-47eaf644982e.

[R52] Mike Gualtieri. Devops is about collaboration, Jun 2011. https://go.forrester.com/blogs/11-06-29-devops_is_about_collaboration_noops_is_about_automation/.

[R53] Team Polyrific. Sdlc series: Noops primer - polyrific, Jun 2018. https://www.polyrific.com/sdlc-series-noops-primer.

[R54] Eveline Oehrlich. What is bizdevops? — devops institute, Sep 2019. https://devopsinstitute.com/what-is-bizdevops/.

[R55] Stephan Thorpe. Cherre, Oct 2020. https://blog.cherre.com/2020/10/12/devops-versus-other-ops/.

[R56] John Allspaw. Reply to http://perfcap.blogspot.com/2012/03/ops ..., Mar 2012. https://gist.github.com/jallspaw/2140086. 26

[R57] Don MacVittie. The role of operations in a no-ops world, Nov 2017. https://devops.com/role-operations-no-ops-world/.

[R58] Kiran Oliver. Why going serverless doesn't mean 'no ops' – the new stack, Jan 2017. https://thenewstack.io/going-serverless-doesnt-mean-no-ops/.

[R59] Joe McKendrick. There's no ops like noops: the next evolution of devops — zdnet, Feb 2019. https://www.zdnet.com/article/theres-no-ops-like-noops-the-next-evolution-of-devops/. 25

[R60] Eshan Sarpotdar. Evolution of noops from devops - cloudhedge, Mar 2019. https://cloudhedge.io/evolution-of-noops-from-devops/.

[R61] Mike Gualtieri. I don't want devops. i want noops. - forrester, Feb 2011. https://go.forrester.com/blogs/11-02-07-i_dont_want_devops_i_want_noops/.

[R62] Zbigniew Cybulski Adam Stempniak. What is aiops, bizdevops, cloudops, devops, itops, noops? a ..., Apr 2019. https://www.stxnext.com/blog/aiops-bizdevops-cloudops-devops-itops-noops-introduction-digital-business-transformation/. 25, 27

[R63] Devi Singh. What is noops and why it is the future of the it industry, Sep 2020. https://insights.daffodilsw.com/blog/what-is-noops-and-why-it-is-the-future-of-the-it-industry.

[R64] Bob Aiello. Devops or noops: Can you have too much automation? — techwell, Nov 2015. https://www.techwell.com/techwell-insights/2015/11/devops-or-noops-can-you-have-too-much-automation.

[R65] Dan Rundle. Are you ready for serverless computing and noops? - worthwhile, Aug 2019. https://worthwhile.com/insights/2019/08/07/serverless-computing-and-noops/. 25

[R66] Alex Melnichuk. A guide to itops, devops, and noops concepts — ncube, Sep 2020. https://ncube.com/blog/itops-devops-and-noops. 25

[R67] Vijay Rayapati. The 2017 cloud trends — from devops to noops — by vijay ..., Feb 2017. https://articles.microservices.com/the-2017-cloud-trends-from-devops-to-noops-1d12fa85d433?gi=749b2d244f1e. 26

[R68] Maha Santaram. Devops to noops a journey worth taking for indian banks - bw cio, May 2019. http://bwcio.businessworld.in/article/DevOps-to-NoOps-A-Journey-worth-taking-for-Indian-Banks/13-05-2019-170428/. 25

[R69] Carten Cordell. Forget devops — is the future of cloud noops? - fedscoop, Jun 2018. https://www.fedscoop.com/forget-devops-future-cloud-government-noops/.

[R70] Derrick Harris. Why 2013 is the year of 'noops' for programmers [infographic ..., Jan 2012. https://gigaom.com/2012/01/31/why-2013-is-the-year-of-noops-for-programmers-infographic/.

[R71] Martin Perlin. No-ops? new-ops? cloud-ops! - evolven, Apr 2012. https://www.evolven.com/blog/noops-newops-cloudops.html.

[R72] Gabriela Motroc. The road to serverless maturity: Running away from "noops" or ..., Feb 2018. https://jaxenter.com/jaxdevops-interview-series-serverless-141273.html.

[R73] Abel Avram. Noops: Its meaning and the debate around it - infoq, Mar 2012. https://www.infoq.com/news/2012/03/NoOps/. 26

[R74] Adrian Bridgwater. Dynatrace gets hands-on with hands-off 'noops' ..., Dec 2019. https://www.forbes.com/sites/adrianbridgwater/2019/12/11/dynatrace-gets-hands-on-with-hands-off-noops-autonomous-cloud/.

[R75] Joe McKendrick. Noops a no-brainer? completely autonomous ..., Feb 2019. https://www.zdnet.com/article/no-ops-like-noops-part-2/.

[R76] Daniele Fontani. Devops trends — towards data science, May 2020. https://towardsdatascience.com/devops-trends-8ccbed85e7af?gi=b3d19c7e664b. 25

[R77] Aaditya Aravamudhan. [webinar recap] noops? or yes, ops! the future ..., Nov 2018. https://blog.opsramp.com/devops-noops.

[R78] Agustin Romano. What is noops?, Jul 2019. https://www.linkedin.com/pulse/what-noops-agustin-romano. 26, 27

[R79] Mike Roberts. Serverless architectures - martin fowler, May 2018. https://www.martinfowler.com/articles/serverless.html.

[R80] Don Dingee. Keeping noops from going rogue, Apr 2019. https://devops.com/keeping-noops-from-going-rogue/.

[R81] Nancy Gohring. Noops' debate grows heated, Mar 2012. https://www.cio.com/article/2398068/-noops--debate-grows-heated.html.

[R82] Margaret Rouse. What is noops?, Jul 2015. https://searchitoperations.techtarget.com/definition/NoOps. 26

[R83] Lewis Prescott. Noops & the future of qa automation (with lewis prescott from ..., Nov 2020. https://theqalead.com/podcast/noops-future-qa-automation-lewis-prescott/.

[R84] Kieron Sambrook-Smith. Drupalcon nashville 2018: Welcome to noops, the new devops ..., Apr 2018. https://www.google.com/url?q=https://www.videodrupal.org/index.php/video/20180412/drupalcon-nashville-2018-welcome-noops-new-devops&sa=U&ved=2ahUKEwiqs5aymr3tAhUHyxoKHZk9Amo4ggEQtwIwBXoECAQQAQ&usg=AOvVaw0WW8vrXfpQ1uszjj7lXTQH.

[R85] Irene Maida. Devops vs noops - criticalcase, Jan 2018. https://www.criticalcase.com/blog/devops-vs-noops.html. 27

[R86] Clive Longbottom. When devops isn't enough, try noops - computer weekly, Jul 2015. https://www.computerweekly.com/opinion/When-DevOps-isnt-enough-try-NoOps. 27

[R87] Meeta Ramnani. Shifting it's focus from operations to outcomes with noops, Jul 2019. https://enterprisetalk.com/featured/shifting-its-focus-from-operations-to-outcomes-with-noops/. 25, 26

[R88] Brightred. [pdf] is noops the end of the road for operations professionals?, Apr 2019. https://www.brightred.com/wp-content/uploads/2019/04/Is-NoOps-the-end-of-the-road-for-operations-professionals.pdf. 21, 27

[R89] Alex Williams. Noops - a big lie or a political shift? - siliconangle, Feb 2012. https://siliconangle.com/2012/02/01/noops-a-big-lie-or-a-political-shift/.

[R90] DIallah Haidar. Oops, netflix: Noops is a no go (nasdaq:nflx, nasdaq:goog ..., Mar 2012. https://www.cheatsheet.com/technology/oops-netflix-noops-is-a-no-go.html/.

[R91] Robyn Weisman. The emotional pull of noops — zenoss, Feb 2018. https://www.zenoss.com/blog/the-emotional-pull-of-noops.

[R92] Natasha Mathur. 2019 deloitte tech trends predictions: Ai-fueled firms, noops ..., Jan 2019. https://hub.packtpub.com/2019-deloitte-tech-trends-predictions-ai-fueled-firms-noops-devsecops-intelligent-interfac

[R93] Vijith Sivadasan. Is noops killing devops- experts disagree - codelattice blog, Dec 2018. https://blog.codelattice.com/is-noops-killing-devops-experts-disagree/.

[R94] Will Kelly. Say 'yes' to noops with the right technology and . . . , Nov 2019. https://searchitoperations.techtarget.com/tip/Say-yes-to-NoOps-with-the-right-technology-and-team-structure. 25, 26

[R95] Crystal Bedell. Cloud computing ushers in a new era for devops: . . . , Jul 2011. https://searchsoftwarequality.techtarget.com/tip/Cloud-computing-ushers-in-a-new-era-for-DevOps-NoOps. 26

[R96] Vishrutha Amudan. Should noops be your new operations strategy? - . . . , Jun 2019. https://medium.com/manageengine/five-worthy-reads-should-noops-be-your-new-operations-strategy-e9b165285620. 25, 26

[R97] Ravindra Yadav. Blog — noops! is that even a term? — gs lab, Apr 2018. https://www.gslab.com/blogs/what-is-noops.

[R98] Ashit Mithal. No-ops – next evolution or fuss? - words geek, Sep 2020. https://wordsgeek.com/tech/ashitmittal/no-ops-next-evolution-or-fuss/.

Appendix A

Search results statistics

Search engine	Search string	# raw results
	No Ops	200
	No Ops culture	20
	No Ops definition	10
	No Ops practices	2
	No Ops challenges	0
	No Ops characteristics	0
	No-Ops	200
	No-Ops culture	20
	No-Ops definition	6
Bing	No-Ops practices	2
	No-Ops challenges	0
	No-Ops characteristics	0
	NoOps	194
	NoOps challenges	9
	NoOps culture	50
	NoOps definition	45
	NoOps practices	30
	NoOps characteristics	0
	No Ops	191
	No Ops culture	0
	No Ops definition	0
	No Ops practices	0
	No Ops challenges	0
	No-Ops	202
	No-Ops culture	23
	No-Ops definition	9
Google	No-Ops practices	7
	No-Ops challenges	0
	No-Ops characteristics	2
	NoOps	200
	NoOps challenges	21
	NoOps characteristics	14
	NoOps culture	55
	NoOps definition	38
	NoOps practices	28
	No Ops	200
	No Ops culture	0

Google Scholar

	No Ops definition	0
	No Ops practices	0
	No Ops challenges	0
	No-Ops	200
	No-Ops culture	0
	No-Ops definition	0
	No-Ops practices	0
	No-Ops challenges	0
	No-Ops characteristics	0
	NoOps	200
	NoOps challenges	0
	NoOps characteristics	0
	NoOps culture	2
	NoOps definition	2
	NoOps practices	0
Total Google	790	
Total Bing	788	
Total Google Scholar	604	
Total	2182	

Appendix B

Preliminary annotation of primary sources

ID	Title	Author	Date	Source type	Publisher type	Job category	Company category
R1	How to get from DevOps to NoOps: 5 steps | The Enterprisers Project	Anita Engleder	13-Mar-2020	Whitepaper	Community	Executive	Product Development
R2	From NoOps to AllOps - Test Double | Our Blog	Jason Grosz	26-Nov-2019	Whitepaper	Company	Engineering	Consultancy
R3	Moving from DevOps to NoOps with a Microservice Architecture on ...	Christopher Hambridge	29-Jun-2016	Whitepaper	Company	Engineering	Product Development
R4	The Evolution of NoOps- Whitepaper on Driving More Value With ...	Mindtree	1-Oct-2020	Whitepaper	Company	Collective authorship	Consultancy
R5	Quality for DevOps teams	Rik Marselis	1-Mar-2020	Book	Book	Executive	Consultancy
R6	No-Ops: How a DevOps strategy centered on automation can ...	Venky Chennapragada	14-May-2020	Whitepaper	Company	Executive	Consultancy
R7	Autonomous Cloud Enablement aka Scaling NoOps via Self-Service ...	Andreas Grabner	7-Feb-2020	Whitepaper	Company	Engineering	Product Development
R8	NoOps, DevSecOps, and Managing Cyber Risk - …	Ken Corless, Mike Kavis, Kieran Norton	27-Feb-2019	Magazine article	Magazine	Executive	Consultancy
R9	Digital transformation: How we took NoOps to the …	Bernd Greifeneder	27-Feb-2020	Whitepaper	Community	Executive	Product Development
R10	wary of the economics of" Serverless" Cloud Computing	Adam Eivy	26-Apr-2017	Paper	Scientific journal	Engineering	Product Development
R11	Introducing a no-ops culture to remove bottlenecks from cloud-native ...	Mike Ensor	14-Jun-2019	Video	Company	Executive	Consultancy
R12	Is NoOps the End of DevOps? Think Again | Blog | AppDynamics	Jordan Bach	11-Apr-2017	Blog article	Company	Engineering	Product Development
R13	NoOps in a serverless world | Deloitte Insights	Ken Corless, Mike Kavis, Kieran Norton	16-Jan-2019	Blog article	Company	Executive	Consultancy
R14	What is NoOps? Is it Agile Ops? - Botmetric	Jayashree Hegde	25-May-2017	Whitepaper	Company	Collective authorship	Product Development
R15	DevOps Ventured, NoOps Gained: The Natural Progression of ...	Harbinder Kang	2-May-2019	Whitepaper	Company	Executive	Product Development
R16	NoOps - Is this the end of DevOps as we know it? - - Stratoscale	Rotem Dafni	31-Oct-2016	Blog article	Company	Executive	Product Development
R17	[PDF] white paper: demystifying noops and serverless computing - CIO.gov	Chief Information Officers Council	1-Jan-2020	Whitepaper	Government	Collective authorship	Governmental organization
R18	NewOps: The Future of Operations in a Hybrid IT World | AVNetwork	Jason Bloomberg	26-Mar-2019	Whitepaper	Community	Executive	Consultancy
R19	What is NoOps and How to Switch from DevOps to NoOps in Mobile ...	Smartface	11-Apr-2018	Whitepaper	Company	Collective authorship	Consultancy
R20	How to Create a Successful NoOps Team | xMatters	Adam Serediuk	12-Nov-2015	Whitepaper	Company	Executive	Product Development
R21	Beware the siren song of no-ops | InfoWorld	David Linthicum	19-Feb-2019	Magazine article	Magazine	Engineering	Publishing medium
R22	The Perspective: NoOps Through the Eyes of a DevOps Engineer	Jayashree Hegde	28-Jan-2017	Blog article	Company	Collective authorship	Product Development
R23	NoOps – The end of DevOps? – KENOPSY	Kenopsy Services	14-Apr-2020	Whitepaper	Company	Engineering	Consultancy
R24	Software Factory #2: What makes infrastructure crucial - Witekio ...	Tangui Colin	1-Sep-2017	Whitepaper	Company	Engineering	Consultancy
R25	framework for managing mission needs, compliance, and trust in the DevOps environment	B.S. Farroha	6-Oct-2014	Paper	Conference	Research	Research
R26	Why culture is more important than tech adoption in a software-driven world	Bernd Greifeneder	20-Jun-2019	Blog article	Community	Executive	Product Development
R27	Redefining NoOps to Better Inform IT Decision-Making	Hussein Badakhchani	30-Aug-2017	Blog article	Community	Executive	Product Development
R28	Serverless Computing: Moving from DevOps to NoOps	Laurent Bride	25-Jun-2018	Blog article	Community	Executive	Consultancy
R29	Logging and NoOps with Christian Beedgen - Software Engineering ...	Christian Beedgen	11-Apr-2016	Podcast	Community	Executive	Product Development
R30	Tech Debate: DevOps vs No Ops - Sysco LABS	Sysco Labs	20-Mar-2018	Video	Company	Collective authorship	Product Development
R33	What is NoOps? The quest for fully automated IT operations | CIO	Mary K. Pratt	11-Jul-2017	Magazine article	Magazine	Content Creation	Publishing medium
R34	How NoOps Improves Development Productivity and Operational ...	Engine Yard	1-Sep-2020	Blog article	Company	Collective authorship	Product Development
R35	Is NoOps Achievable? - Lumen	Scott Brindamour	15-Jul-2020	Blog article	Company	Engineering	Product Development
R36	A NoOps state of mind - SD Times	Christina Cardoza	13-Mar-2019	Magazine article	Magazine	Content Creation	Publishing medium
R37	NoOps: How serverless architecture introduces a third mode of IT ...	Keith Townsend	19-Jan-2018	Blog article	Magazine	Engineering	Consultancy
R38	NoOps: Importance and its Benefits in IT Operations | E-SPIN Group	E-Spincorp	28-Mar-2019	Blog article	Company	Collective authorship	Consultancy
R39	Look mum, NoOps! How to empower the next evolution of IT ...	Michael Allen	5-May-2020	Blog article	Community	Executive	Product Development
R40	From DevOps to NoOps – how to automate your chocolate factory	Ruben van der Zwan	7-May-2017	Blog article	Company	Executive	Consultancy
R41	Your Cloud Management Team: Ops, DevOps or NoOps ? - IOD	Ofir Nachmani	26-Dec-2011	Blog article	Magazine	Executive	Publishing medium
R42	Ops, DevOps and PaaS (NoOps) at Netflix - Adrian Cockcroft's Blog	Adrian Cockcroft	19-Mar-2012	Blog article	Ad-hoc blog	Executive	Product Development
R43	The Road to NoOps: Serverless Computing is Quickly Gaining ...	Mark Boyd	18-May-2016	Magazine article	Magazine	Content Creation	Publishing medium
R44	DevOps, DataOps, GitOps, NoOps ... let's clarify! - Mia Platform	Mia-Platform Team	12-Feb-2020	Blog article	Company	Collective authorship	Product Development
R45	NoOps, AppOps, DevOps, & More - Removing the OS Barrier with ...	Adron Hall	8-Feb-2012	Blog article	Company	Engineering	Product Development
R46	NoOps – Reality or just another fad | Brillio Technologies	Siva Perubotla	12-Mar-2019	Blog article	Company	Executive	Consultancy
R47	No-Ops: It's Been 5 Years - DevOps.com	Don Macvittie	1-Dec-2017	Blog article	Community	Marketing	Consultancy

ID	Title	Author	Date	Source type	Publisher type	Job category	Company category
R49	The blueprints of a "no-ops" startup | Hacker Noon	Tal Bereznitskey	9-Aug-2017	Blog article	Community	Executive	Product Development
R50	Netflix uses lots of cloud services -- but don't call it 'NoOps ...	Ellen Messmer	27-Mar-2012	Magazine article	Magazine	Content Creation	Publishing medium
R51	NoOps is a No-No. In my previous post about the evolution… | by ...	Colin But	21-May-2018	Blog article	Community	Engineering	Product Development
R52	DevOps Is About Collaboration; NoOps Is About …	Mike Gualtieri	29-Jun-2011	Blog article	Company	Research	Research
R53	SDLC Series: NoOps Primer - Polyrific	Team Polyrific	2-Jun-2018	Blog article	Company	Collective authorship	Consultancy
R54	What is BizDevOps? — DevOps Institute	Eveline Oehrlich	9-Sep-2019	Blog article	Company	Marketing	Publishing medium
R55	Cherre	Stephan Thorpe	12-Oct-2020	Blog article	Company	Executive	Product Development
R56	Reply to http://perfcap.blogspot.com/2012/03/ops …	John Allspaw	20-Mar-2012	Blog article	Community	Executive	Product Development
R57	The Role of Operations in a No-Ops World	Don MacVittie	28-Nov-2017	Blog article	Community	Marketing	Consultancy
R58	Why Going Serverless Doesn't Mean 'No Ops' – The New Stack	Kiran Oliver	20-Jan-2017	Podcast	Magazine	Content Creation	Publishing medium
R59	There's no ops like NoOps: the next evolution of DevOps | ZDNet	Joe McKendrick	23-Feb-2019	Blog article	Community	Engineering	Publishing medium
R60	Evolution of NoOps from DevOps - CloudHedge	Eshan Sarpotdar	19-Mar-2019	Blog article	Company	Executive	Consultancy
R61	I Don't Want DevOps. I Want NoOps. - Forrester	Mike Gualtieri	8-Feb-2011	Blog article	Company	Research	Research
R62	What Is AIOps, BizDevOps, CloudOps, DevOps, ITOps, NoOps? A ...	Adam Stempniak, Zbigniew Cybulski	25-Apr-2019	Blog article	Company	Content Creation	Consultancy
R63	What is NoOps and Why it is the Future of the IT Industry	Devi Singh	15-Sep-2020	Blog article	Company	Content Creation	Consultancy
R64	DevOps or NoOps: Can You Have Too Much Automation? | TechWell	Bob Aiello	19-Nov-2015	Blog article	Community	Executive	Consultancy
R65	Are You Ready for Serverless Computing and NoOps? - Worthwhile	Dan Rundle	22-Aug-2019	Blog article	Company	Executive	Consultancy
R66	A Guide to ITOps, DevOps, and NoOps Concepts | NCube	Alex Melnichuk	25-Sep-2020	Whitepaper	Company	Marketing	Consultancy
R67	The 2017 Cloud Trends — From DevOps to NoOps | by Vijay ...	Vijay Rayapati	8-Feb-2017	Blog article	Community	Executive	Product Development
R68	DevOps to NoOps A Journey worth taking for Indian Banks - BW CIO	Maha Santaram	13-May-2019	Blog article	Magazine	Engineering	Consultancy
R69	Forget DevOps — Is the future of cloud NoOps? - FedScoop	Carten Cordell	29-Jun-2018	Magazine article	Magazine	Content Creation	Publishing medium
R70	Why 2013 is the year of 'NoOps' for programmers [Infographic ...	Derrick Harris	31-Jan-2012	Blog article	Company	Content Creation	Research
R71	No-Ops? New-Ops? Cloud-Ops! - Evolven	Martin Perlin	3-Apr-2012	Blog article	Company	Executive	Product Development
R72	The road to serverless maturity: Running away from "NoOps" or ...	Gabriela Motroc	12-Feb-2018	Magazine article	Magazine	Content Creation	Publishing medium
R73	NoOps: Its Meaning and the Debate around It - InfoQ	Abel Avram	16-Mar-2012	Blog article	Community	Content Creation	Publishing medium
R74	Dynatrace Gets Hands-On With Hands-Off 'NoOps' …	Adrian Bridgwater	11-Dec-2019	Magazine article	Magazine	Content Creation	Publishing medium
R75	NoOps a no-brainer? Completely autonomous …	Joe McKendrick	27-Feb-2019	Blog article	Community	Engineering	Publishing medium
R76	DevOps Trends | Towards Data Science	Daniele Fontani	20-May-2020	Blog article	Community	Executive	Consultancy
R77	[Webinar Recap] NoOps? Or Yes, Ops! The Future …	Aaditya Aravamudhan	21-Nov-2018	Blog article	Company	Content Creation	Product Development
R78	What Is NoOps?	Agustin Romano	19-Jul-2019	Blog article	Community	Engineering	Consultancy
R79	Serverless Architectures - Martin Fowler	Mike Roberts	22-May-2018	Blog article	Ad-hoc blog	Executive	Consultancy
R80	Keeping NoOps from going rogue	Don Dingee	29-Apr-2019	Blog article	Community	Content Creation	Marketing
R81	NoOps' Debate Grows Heated	Nancy Gohring	20-Mar-2012	Magazine article	Magazine	Content Creation	Publishing medium
R82	What is NoOps?	Margaret Rouse	1-Jul-2015	Blog article	Community	Content Creation	Publishing medium
R85	DevOps vs NoOps - Criticalcase	Irene Maida	18-Jan-2018	Blog article	Company	Content Creation	Consultancy
R86	When DevOps isn't enough, try NoOps - Computer Weekly	Clive Longbottom	27-Jul-2015	Blog article	Community	Research	Research
R87	Shifting It's Focus from Operations to Outcomes With NoOps	Meeta Ramnani	17-Jul-2019	Blog article	Community	Content Creation	Publishing medium
R88	[PDF] Is NoOps the end of the road for operations professionals?	Brightred	5-Apr-2019	Whitepaper	Company	Collective authorship	Recruitment
R89	NoOps - A Big Lie or a Political Shift? - SiliconANGLE	Alex Williams	1-Feb-2012	Blog article	Magazine	Content Creation	Publishing medium
R90	Oops, Netflix: NoOps is a No Go (NASDAQ:NFLX, NASDAQ:GOOG ...	DIallah Haidar	28 March 2012	Magazine article	Magazine	Content Creation	Publishing medium
R91	The Emotional Pull of NoOps | Zenoss	Robyn Weisman	28-Feb-2018	Blog article	Company	Content Creation	Product Development
R92	2019 Deloitte tech trends predictions: AI-fueled firms, NoOps ...	Natasha Mathur	29-Jan-2019	Blog article	Community	Content Creation	Publishing medium
R93	Is NoOps Killing DevOps- Experts Disagree - Codelattice Blog	Vijith Sivadasan	5-Dec-2018	Blog article	Company	Executive	Consultancy
R94	Say 'yes' to NoOps with the right technology and …	Will Kelly	26-Nov-2019	Blog article	Community	Content Creation	Publishing medium
R95	Cloud computing ushers in a new era for DevOps: …	Crystal Bedell	11-Jul-2011	Blog article	Community	Content Creation	Publishing medium
R96	Should NoOps be your new operations strategy? - …	Vishrutha Amudan	24-Jun-2019	Blog article	Community	Marketing	Consultancy
R97	Blog | NoOps! Is That Even A Term? | GS Lab	Ravindra Yadav	24-Apr-2018	Blog article	Company	Executive	Consultancy
R98	No-Ops – Next Evolution or Fuss? - Words Geek	Ashit Mithal	20-Sep-2020	Blog article	Ad-hoc blog	Engineering	Publishing medium

Appendix C

Mapping of primary sources to findings

Code	Relevant primary studies
NoOps implies that an IT environment is very automated, such that there is no necessity for a dedicated team to manage operations in-house	R1, R33, R36, R50, R55, R64, R78,
NoOps is the goal of completely automating the deployment, monitoring, management of applications and infrastructure	R22, R44, R51, R54, R63, R69, R75, R78, R82, R86, R95, R97,
NoOps implies that an IT environment is very automated and abstracted from the underlying infrastructure, such that there is a reduced necessity for a dedicated team to manage operations in-house	R13, R15, R77, R92,
NoOps implies that an IT environment is very automated, such that there is a reduced necessity for a dedicated team to manage operations in-house	R5, R8, R9, R16,
NoOps implies that developers do not have to interact with the operations team	R12, R14, R19, R33, R38, R50, R53, R61, R71, R88, R90, R93, R97,
NoOps implies that an IT environment is very automated and abstracted from the underlying infrastructure, such that there no necessity for a dedicated team to manage operations in-house	R16, R18, R21, R22, R24, R34, R59, R82,
NoOps is the mindset that no traditional operational tasks are necessary to deploy and operate software	R7, R17, R26, R27, R33, R36, R42, R44, R53, R59, R60, R66, R94,
NoOps implies total exclusion of IT Ops	R3, R25, R27, R28, R29, R30, R57, R62, R76, R96, R98,
NoOps is an augmentation of DevOps	R3, R4, R11, R12, R13, R14, R17, R19, R22, R23, R27, R36, R52, R85, R94,
Provides scalability	R1, R65, R87, R94,
Facilitates testing automation	R4,
Lowers the risk for cyberattacks	R8,
Reduces necessity for scaling ITOps team size	R9, R33, R39, R81,
Improves collaboration	R9, R95,
Enables self-service	R11,

Increases accountability	R11,
Build in Governance, Observability, Quality and Compliance	R11,
Enhances operations process with further intelligence, reducing the need for incident responses	R3, R8, R38,
Facilitates quality engineering	R7,
Enhances reliability	R68,
Incentivizes developers to write better quality code	R29,
Minimizes the risk of human error	R5, R11, R40, R62, R66, R93,
More time and opportunities for learning and growth for ITOps team	R13, R14, R23, R29, R33, R36, R62, R81, R96,
Cost savings for infrastructure and application operations	R4, R6, R13, R17, R30, R34, R46, R63, R65, R68, R87, R96,
Additional operational efficiency for infrastructure and application operations	R9, R14, R15, R17, R34, R59, R62, R65, R66, R68, R76, R93,
Eliminates friction with the infrastructure operations (especially in the release phase), such that developers can create value by not having to interoperate with operations teams	R4, R11, R12, R13, R28, R30, R33, R35, R39, R55, R59, R63, R65, R66, R69, R88, R93, R96,
Cost-benefit analysis for deciding which aspects to automate	R2, R52,
Quality engineering	R4,
Virtual assistants	R6,
Support for Site Reliability Engineering	R11,
Orchestration mechanisms	R11, R33, R78, R94, R96,
Microservice architecture suitability	R14, R52, R67, R95,
Chaos engineering	R30,
Automated security scans	R13, R27, R52, R76,
Automation of test-release-operate phases	R4, R14, R42, R73,
Automated testing	R4,
Automated reporting of critical information	R3, R19,
Software bots	R6,
Continuous Delivery	R6, R20, R39,
Ensuring service observability	R11,
Continuous Integration tooling	R11, R67, R85, R94,
Build automation	R4,
Self-diagnostics	R1, R13,
Auto-scaling	R3, R14, R34,
Automated management of applications	R14, R16, R40, R53, R88, R95,
Automated infrastructure provisioning	R13, R33, R52, R96,
Implementing feedback cycles	R2, R3, R26,
Self-service portals for operations	R6, R7, R14, R56, R77, R94,
AI/ML-based automation of operations	R1, R4, R15, R35, R46, R86, R94,
Proactive monitoring	R6, R7, R18, R46, R52,
Self-healing	R3, R7, R9, R13, R39, R67, R74,

Everything-as-Code	R11, R13, R23, R24, R73, R76,
Automated deployment	R1, R6, R11, R23, R30, R38, R42,
Automated monitoring	R3, R11, R12, R14, R15, R20, R23, R34, R42, R47, R52, R60, R67,
Outsourcing operational activities to serverless/cloud providers	R8, R11, R13, R14, R15, R16, R17, R18, R19, R21, R22, R23, R25, R26, R27, R28, R30, R35, R36, R37, R41, R42, R43, R44, R45, R49, R50, R52, R55, R57, R59, R60, R62, R63, R65, R67, R69, R70, R71, R72, R77, R78, R79, R80, R81, R82, R85, R86, R87, R89, R90, R92, R94, R95, R96, R98,
Aiming for process reproducibility	R2,
Proactivity in engineering the variability of operations	R13, R88,
Removing the walls between IT and business	R13, R94, R95,
Shifting developers' thoughts towards taking ownership of the platforms	R23,
Shifting responsibility for operations from ITOps teams to Software Architects	R26,
Shifting monitoring responsibility from the ITOps teams to the development teams	R1, R6,
Shifting security operations from production stage to development stage	R8, R13,
Training teams on new automation tools	R9, R13,
Transfer of skills	R15, R17, R26, R41, R77,
Shifting quality assurance from production stage to development stage	R1, R6,
Management involvement and acceptance	R1, R13,
Removing the silo between Dev and ITOps teams	R3, R9, R11, R13, R16, R27, R39, R47, R77, R89, R95,
Responsibility for deployment and operation is shifted to software engineering teams	R9, R11, R14, R20, R30, R82, R93,
NoOps can not be based on traditional ways of working together	R9,
Reported hesitation to understanding the business in the initial phases of adoption	R9, R26,
Reported discomfort in the initial phases of adoption	R1, R9, R13, R85, R88,
Management of functions when using FaaS solutions	R17,
Poor NoOps implementation quality could lead to cyber risk at scale	R8, R13, R65,
Vendor lock-in	R13, R17, R65, R80,
Outsourcing activities to third-parties could lead to technical issues and limitations	R2, R13, R37, R62, R78, R80, R96,

Adoption may not be suitable or requires re-engineering of legacy systems	R11, R12, R13, R15, R17, R18, R19, R23, R30, R33, R34, R44, R54, R57, R59, R77, R78, R85, R88, R93, R98,
NoOps requires a mindset shift for developers in order to ensure that the developed artifact behaves the same in both local and production environments	R16, R29, R64, R78, R86, R98,
Regulatory compliance, security and other cross-cutting concerns need to be addressed company-wide	R11, R23, R24,
Adopting NoOps could lead to less involvement from development in operations, which completely contradicts DevOps practices	R16, R25, R39,

Appendix D

Tabular representation of the qualitative findings

Code	# paragraphs	# sources
NoOps implies that an IT environment is highly automated and abstracted from the underlying infrastructure, such that there is a reduced necessity for a dedicated team to manage operations in-house	4	4
NoOps implies that an IT environment is highly automated, such that there is a reduced necessity for a dedicated team to manage operations in-house	4	4
NoOps implies that an IT environment is highly automated, such that there is no necessity for a dedicated team to manage operations in-house	7	7
NoOps implies that an IT environment is highly automated and abstracted from the underlying infrastructure, such that there no necessity for a dedicated team to manage operations in-house	8	8
NoOps implies total exclusion of IT Ops	11	11
NoOps is the goal of completely automating the deployment, monitoring, management of applications and infrastructure	12	12
NoOps is the mindset that no traditional operational tasks are necessary to deploy and operate software	14	13
NoOps implies that developers do not have to interact with the operations team	15	13

Table D.1: The frequency of codes related to the NoOps definition

Code	**# paragraphs**	**# sources**
Facilitates testing automation	1	1
Lowers the risk for cyberattacks	1	1
Enables self-service	1	1
Increases accountability	1	1
Build in Governance, Observability, Quality and Compliance	1	1
Improves collaboration	2	2
Facilitates quality engineering	2	1
Enhances operations process with further intelligence, reducing the need for incident responses	3	3
Enhances reliability	3	1
Incentivizes developers to write better quality code	2	1
Provides scalability	4	4
Reduces necessity for scaling ITOps team size	5	4
Minimizes the risk of human error	8	6
More time and opportunities for learning and growth for ITOps team	9	9
Cost savings for infrastructure and application operations	14	12
Additional operational efficiency for infrastructure and application operations	17	12
Eliminates friction with the infrastructure operations (especially in the release phase), such that developers can create value by not having to interoperate with operations teams	22	18

Table D.2: The frequency of codes related to the NoOps advantages

Code	**# paragraphs**	**# sources**
Quality engineering	1	1
Virtual assistants	1	1
Support for Site Reliability Engineering	1	1
Chaos engineering	1	1
Cost-benefit analysis for deciding which aspects to automate	2	2
Automated testing	2	1
Automated reporting of critical information	2	2
Software bots	2	1
Ensuring service observability	2	1
Continuous Delivery	3	3
Build automation	3	1
Self-diagnostics	3	2
Automated security scans	4	4
Implementing feedback cycles	4	3
Orchestration mechanisms	5	5
Microservice architecture suitability	5	4
Automation of test-release-operate phases	5	4
Auto-scaling	6	3
Automated management of applications	7	6
Proactive monitoring	7	5
Continuous Integration tooling	8	4
Self-service portals for operations	8	6
Automated infrastructure provisioning	9	4
Self-healing	9	7
AI/ML-based automation of operations	11	7
Everything-as-Code	11	6
Automated deployment	14	7
Automated monitoring	23	13
Outsourcing operational activities to serverless/cloud providers	85	56

Table D.3: The frequency of codes related to the NoOps technical characteristics

Code	# paragraphs	# sources
Aiming for process reproducibility	1	1
Shifting developers' thoughts towards taking ownership of the platforms	1	1
Shifting responsibility for operations from ITOps teams to Software Architects	1	1
Proactivity in engineering the variability of operations	2	2
Shifting monitoring responsibility from the ITOps teams to the development teams	2	2
Shifting security operations from production stage to development stage	2	2
Training teams on new automation tools	2	2
Removing the walls between IT and business	3	3
Shifting quality assurance from production stage to development stage	3	2
Management involvement and acceptance	3	2
Transfer of skills	5	5
Responsibility for deployment and operation is shifted to software engineering teams	10	7
Removing the silo between Dev and ITOps teams	13	11

Table D.4: The frequency of codes related to the NoOps cultural characteristics

Code	# paragraphs	# sources
NoOps can not be based on traditional ways of working together	1	1
Reported hesitation to understanding the business in the initial phases of adoption	2	2
Reported discomfort in the initial phases of adoption	5	5

Table D.5: The frequency of codes related to the NoOps social challenges

Code	# paragraphs	# sources
Management of functions when using FaaS solutions	1	1
Poor NoOps implementation quality could lead to cyber risk at scale	3	3
Vendor lock-in	4	4
Outsourcing activities to third-parties could lead to technical issues and limitations	11	7
Adoption may not be suitable or requires re-engineering of legacy systems	30	21

Table D.6: The frequency of codes related to the NoOps technical challenges

Code	# paragraphs	# sources
NoOps requires a mindset shift for developers in order to ensure that the developed artifact behaves the same in both local and production environments	6	6

Table D.7: The frequency of codes related to the NoOps cultural challenges

Code	# paragraphs	# sources
Regulatory compliance, security and other cross-cutting concerns need to be addressed company-wide	3	3

Table D.8: The frequency of codes related to the NoOps organizational challenges

Code	# paragraphs	# sources
Adopting NoOps could lead to less involvement from development in operations, which completely contradicts DevOps practices	3	3

Table D.9: The frequency of codes related to the NoOps general challenges

Appendix E

Mappings between job expertise or company domains and study findings

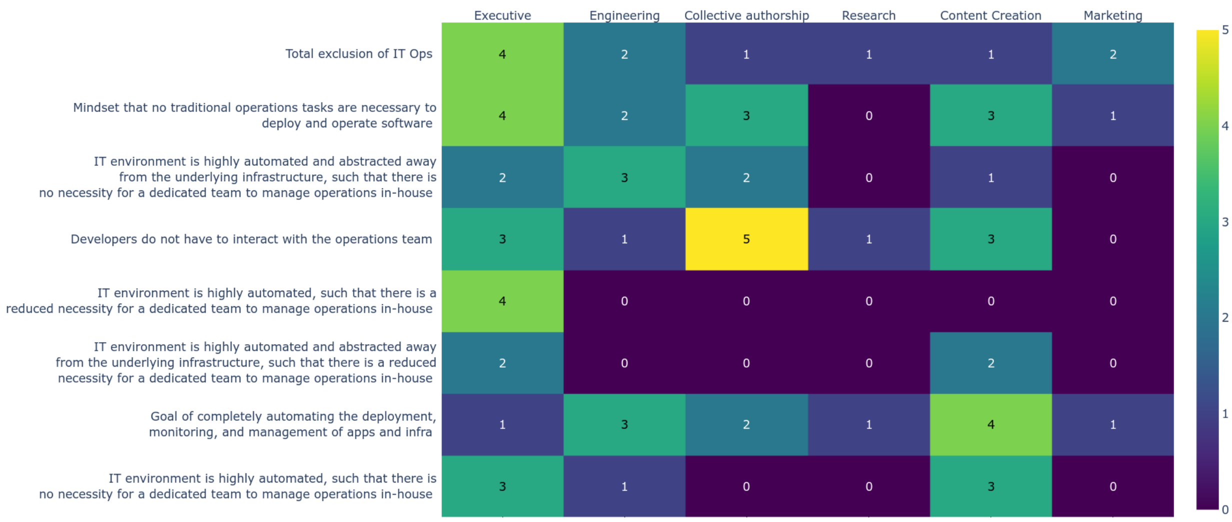


Figure E.1: Heatmap showing the mapping between the authors' job types and the reported NoOps definitions

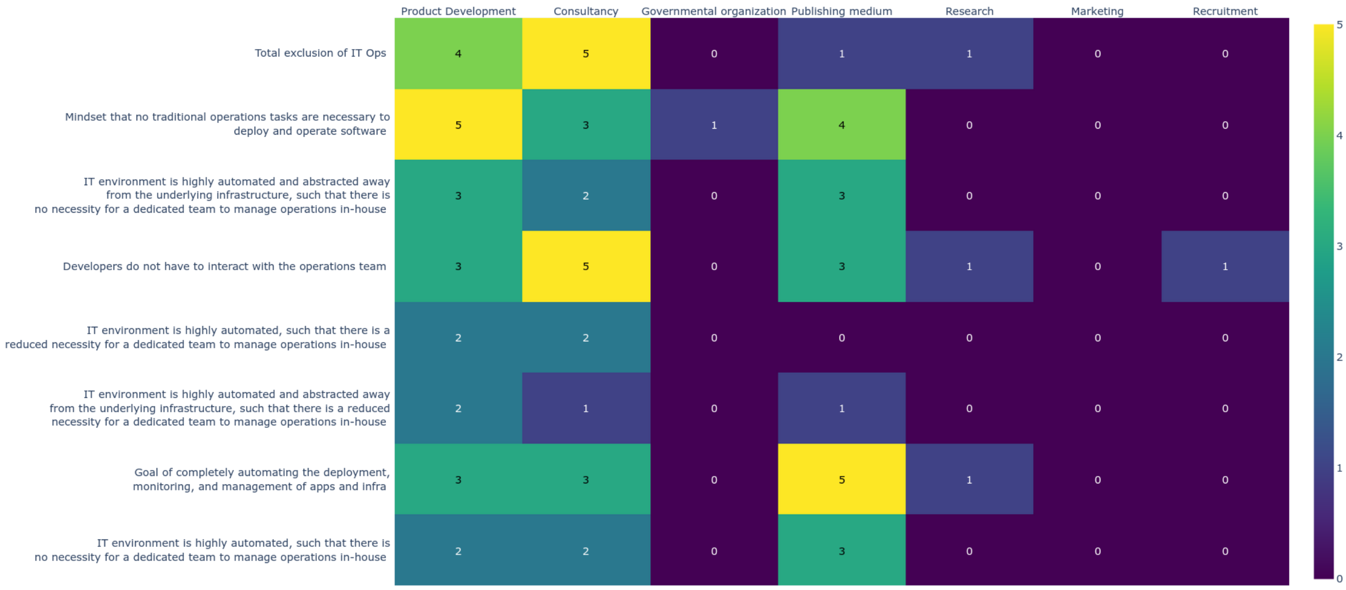


Figure E.2: Heatmap showing the mapping between the types of companies represented in the entire pool of sources and the reported NoOps definitions

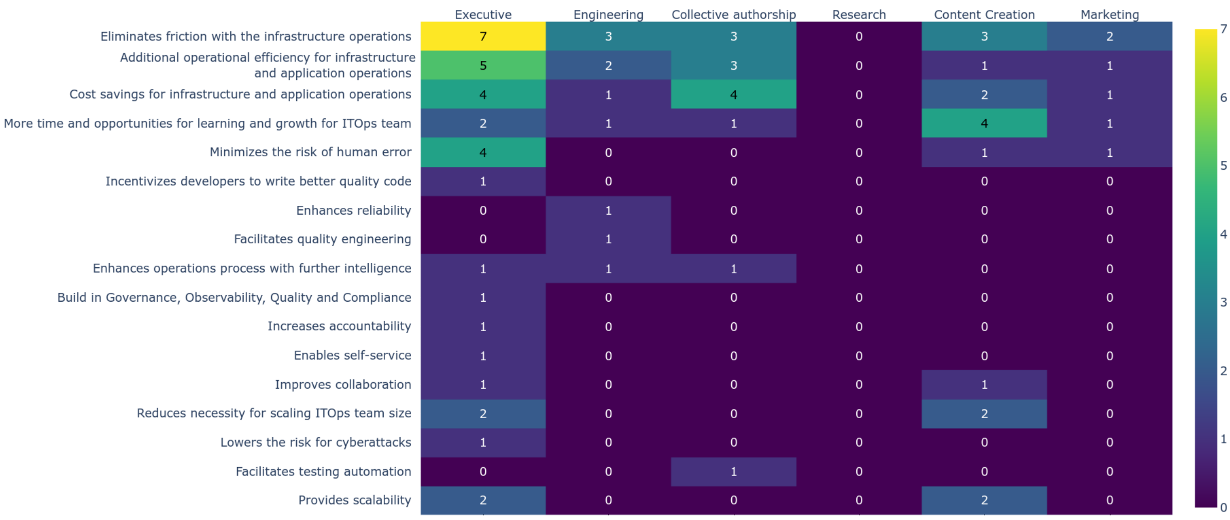


Figure E.3: Heatmap showing the mapping between the authors' job types and the reported NoOps advantages

Figure E.4: Heatmap showing the mapping between the types of companies represented in the entire pool of sources and the reported NoOps advantages

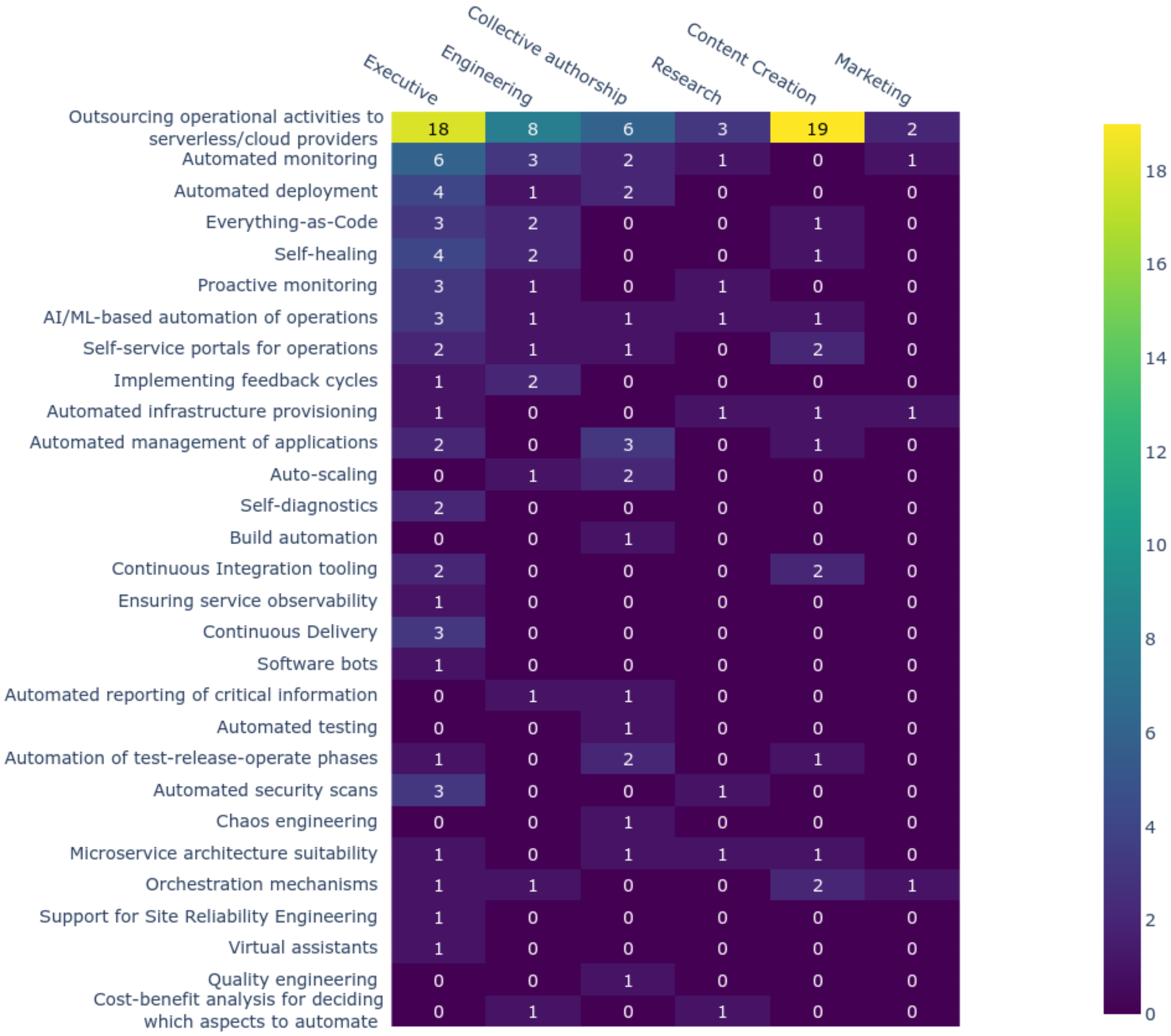


Figure E.5: Heatmap showing the mapping between the authors' job types and the reported NoOps technical characteristics

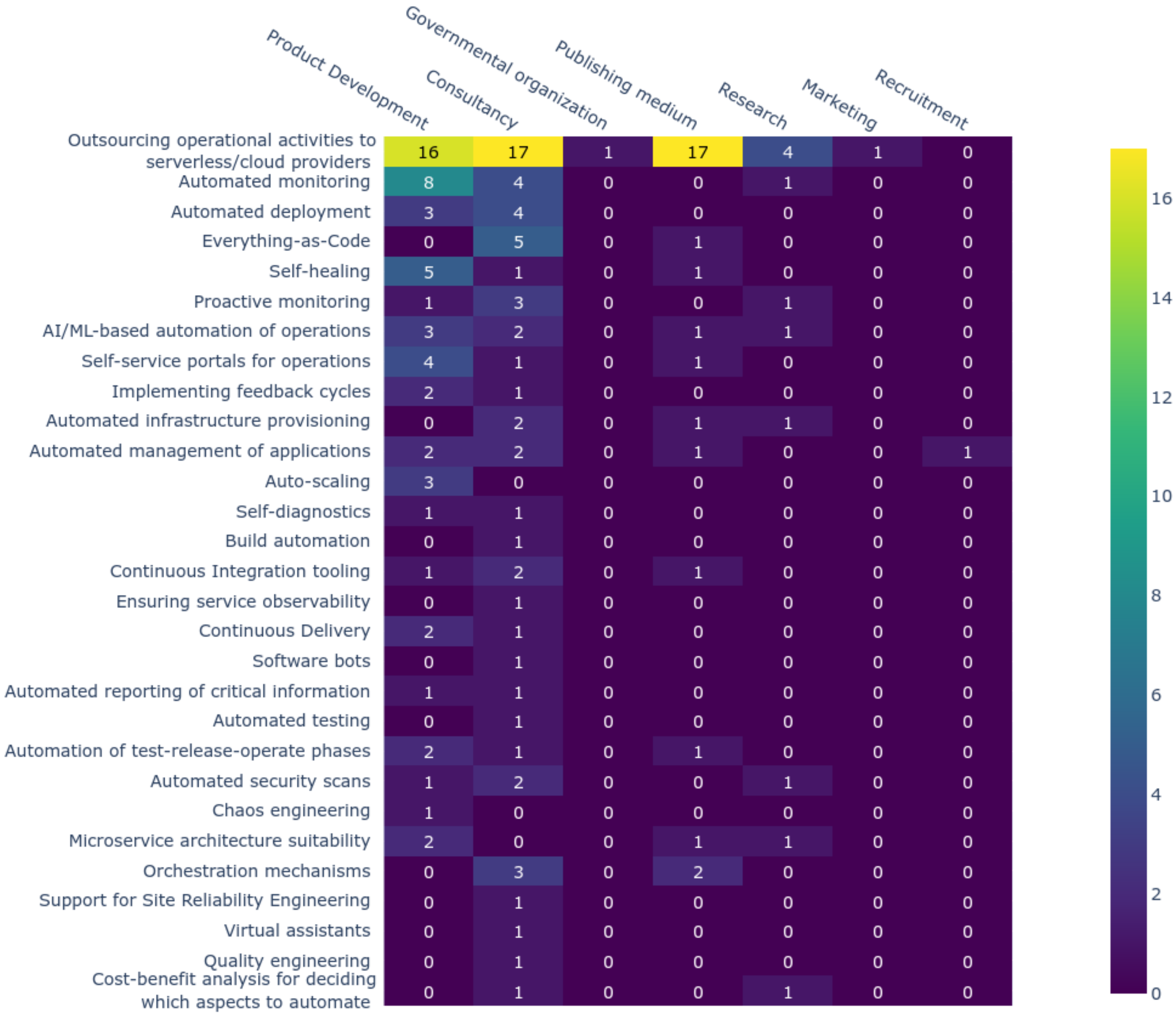


Figure E.6: Heatmap showing the mapping between the types of companies represented in the entire pool of sources and the reported NoOps technical characteristics

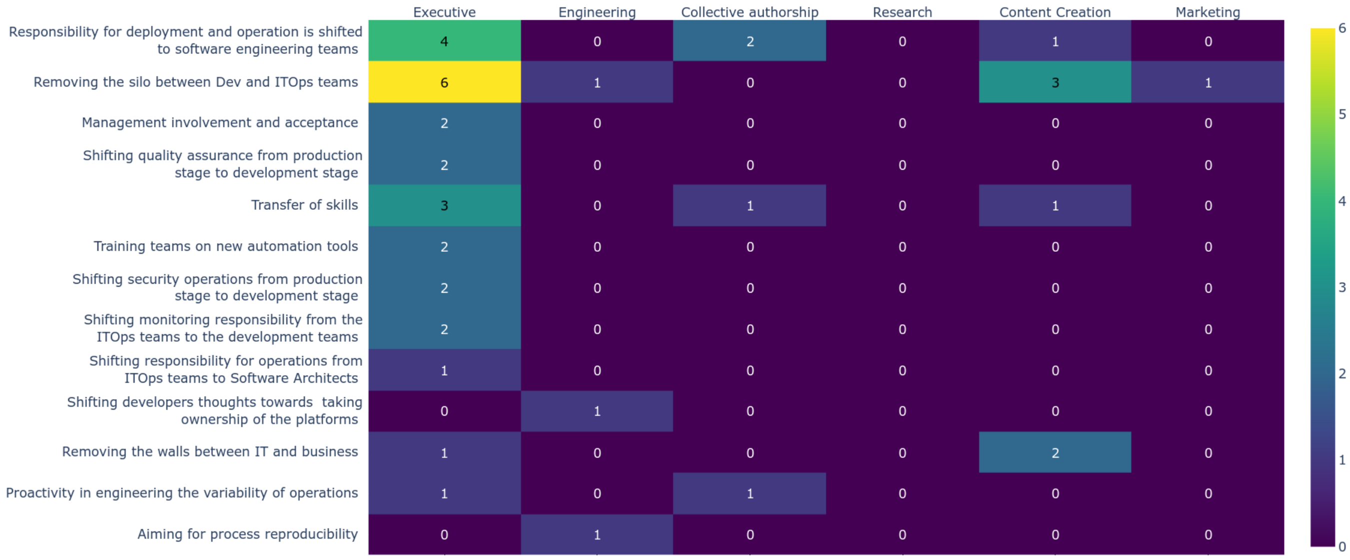


Figure E.7: Heatmap showing the mapping between the authors' job types and the reported NoOps cultural characteristics

Figure E.8: Heatmap showing the mapping between the types of companies represented in the entire pool of sources and the reported NoOps cultural characteristics

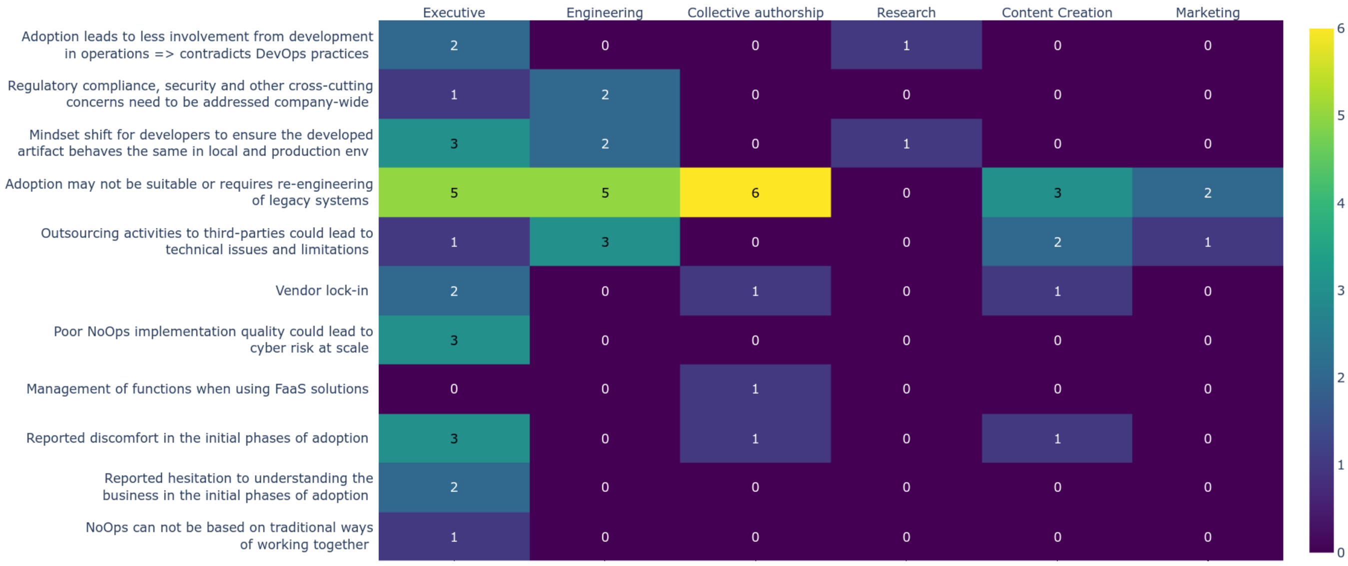


Figure E.9: Heatmap showing the mapping between the authors' job types and the reported NoOps challenges

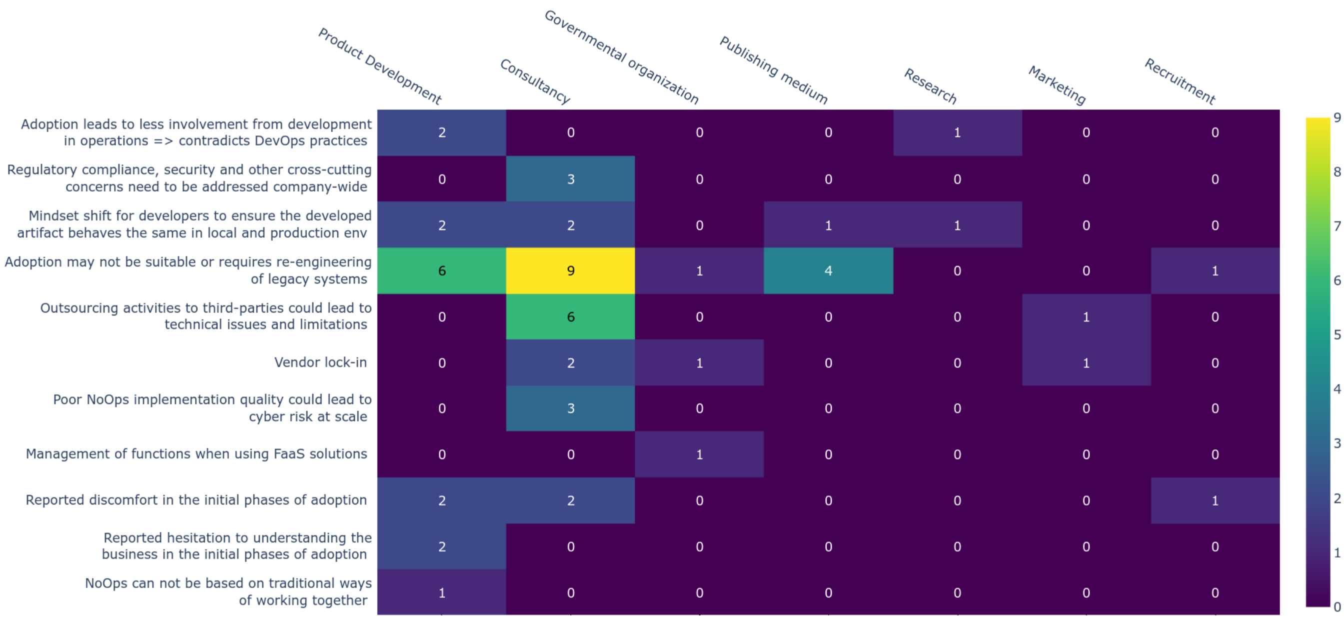
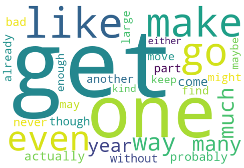


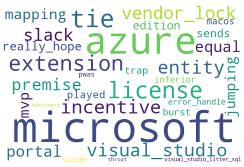
Figure E.10: Heatmap showing the mapping between the types of companies represented in the entire pool of sources and the reported NoOps challenges

Appendix F

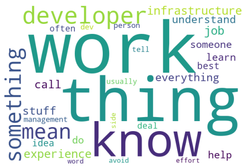
Topic modeling wordclouds



(a) Topic 0

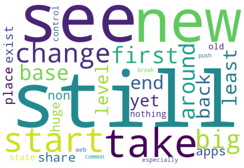


(b) Topic 1



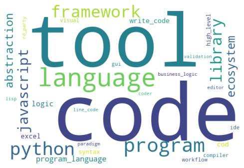
(c) Topic 2

(d) Topic 3



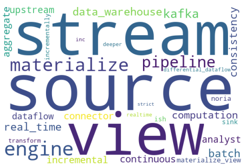
(e) Topic 4

(f) Topic 5

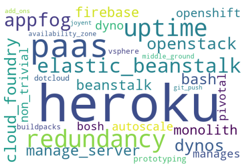


(g) Topic 6

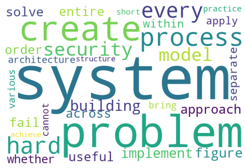
(h) Topic 7



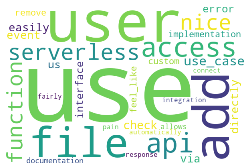
(i) Topic 8



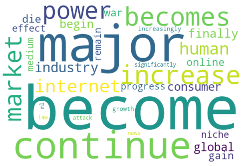
(j) Topic 9



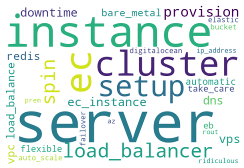
(k) Topic 10



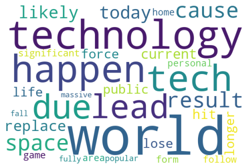
(l) Topic 11



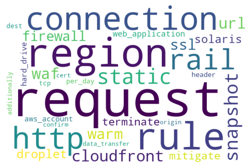
(m) Topic 12



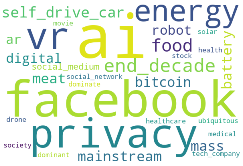
(n) Topic 13



(o) Topic 14



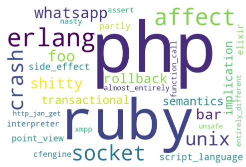
(p) Topic 15



(q) Topic 16

(r) Topic 17

(s) Topic 18



(t) Topic 19

Figure F.1: Word clouds for topics 0 to 19

(u) Topic 20 (v) Topic 21 (w) Topic 22 (x) Topic 23

(y) Topic 24 (z) Topic 25 (aa) Topic 26 (ab) Topic 27

(ac) Topic 28 (ad) Topic 29 (ae) Topic 30 (af) Topic 31

(ag) Topic 32 (ah) Topic 33 (ai) Topic 34 (aj) Topic 35

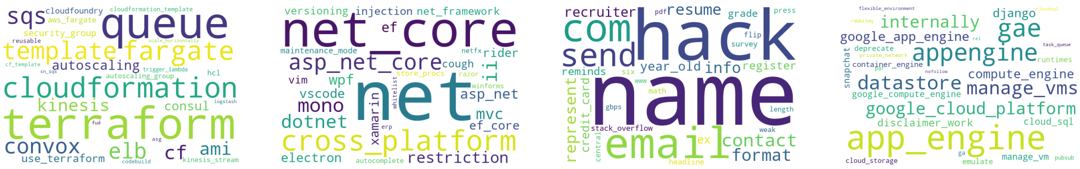
(ak) Topic 36 (al) Topic 37 (am) Topic 38 (an) Topic 39

(ao) Topic 40 (ap) Topic 41 (aq) Topic 42 (ar) Topic 43

(as) Topic 44 (at) Topic 45 (au) Topic 46 (av) Topic 47

(aw) Topic 48 (ax) Topic 49 (ay) Topic 50 (az) Topic 51

Figure F.1: Word clouds for topics 20 to 51



(ba) Topic 52 (bb) Topic 53 (bc) Topic 54 (bd) Topic 55

Figure F.1: Word clouds for topics 52 to 55

Appendix G

Interpretation of topic modeling results

Topic ID	Words	Interpretation	Mapping codes
0	get, one, like, make, go, even, way, much, many, year, actually, come, probably, without, though, part, never, bad, move, another, may, might, find, already, large, enough, maybe, keep, kind, either	Not enough information for interpretation. Seems to be a collection of words used in many comments, but without a lot of standalone semantic value.	N/A
1	microsoft, azure, tie, license, extension, incentive, entity, visual_studio, vendor_lock, slack, premise, portal, equal, mapping, mvp, jumping, really_hope, edition, sends, played, trap, burst, macos, inferior, error_handle, silver, pwas, visual_studio_litter_sql, throat, machinery	Microsoft Azure is a cloud service provider. There is also the challenge of vendor lock-in, showing that in this discussion there was probably a discussion about the challenges of Azure.	Outsourcing operational activities to serverless/cloud providers, Vendor lock-in
2	work, thing, know, developer, something, mean, infrastructure, experience, job, stuff, call, everything, understand, do, learn, help, someone, idea, best, deal, often, dev, person, management, tell, usually, avoid, side, word, effort	Probably related to developers working on infrastructure operations. Not enough information for mapping to existing codes.	N/A

3	would, well, time, also, lot, really, want, support, could, point, give, issue, great, project, since, feature, sure, two, reason, platform, day, provide, case, bit, pretty, far, let, compare, course, option	Not enough information for interpretation	N/A
4	still, see, new, take, start, change, big, first, around, base, least, end, back, level, yet, apps, share, non, exist, place, huge, nothing, old, control, state, especially, web, common, push, break	Not enough information for interpretation	N/A
5	think, say, good, try, look, seem, right, anything, always, talk, agree, put, yes, interest, post, article, mention, ever, comment, believe, feel, thought, guess, wrong, last, ask, exactly, edit, guy, question	Not enough information for interpretation	N/A
6	code, tool, language, program, python, framework, library, javascript, ecosystem, abstraction, program_language, logic, cod, write_code, excel, syntax, high_level, visual, gui, compiler, ide, workflow, paradigm, rd_party, editor, business_logic, line_code, validation, lisp, coder	Related to programming languages, tools and frameworks. Not related specifically to NoOps.	N/A
7	people, company, product, less, business, high, startup, low, value, money, stop, sell, buy, grow, demand, paid, rate, risk, reduce, quality, chance, cut, towards, sense, outsource, house, competitor, netflix, employee, history	Business-related topic, probably about the role of people in startups, product development and outsourcing activities. Not related directly to NoOps.	N/A

8	source, stream, view, engine, materialize, pipeline, kafka, real_time, batch, data_warehouse, consistency, computation, continuous, aggregate, connector, incremental, upstream, analyst, dataflow, materialize_view, noria, sink, ish, transform, incrementally, differential_dataflow, deeper, realtime, inc, strict	Related to Big Data storage (data warehouses) and processing methods (Kafka streams, view, dataflow, transform, materialize_view, computation, aggregations).	
9	heroku, paas, uptime, redundancy, elastic_beanstalk, appfog, openstack, dynos, cloud_foundry, manage_server, bash, beanstalk, firebase, monolith, openshift, dyno, non_trivial, autoscale, manages, pivotal, bosh, prototyping, dotcloud, buildpacks, vsphere, availability_zone, middle_ground, joyent, add_ons, git_push	Multiple cloud-related services are discussed (Heroku, Elastic Beanstalk, AppFog, OpenStack, CloudFoundry, Firebase, OpenShift). Autoscaling is mentioned as well. We see some cloud-specific terms as well, such as dynos (Heroku-specific containers) and availability zones.	Outsourcing operational activities to serverless/cloud providers, Auto-scaling
10	system, problem, create, process, hard, every, security, model, building, implement, figure, approach, solve, fail, entire, useful, order, across, whether, separate, within, apply, architecture, various, cannot, bring, practice, short, structure, achieve	Probably about the management of system problems and security. Not enough information.	N/A
11	use, user, add, file, access, api, serverless, nice, function, use_case, check, via, directly, easily, event, us, error, interface, implementation, remove, custom, feel_like, documentation, allows, automatically, response, integration, connect, pain, fairly	We see a clear mention of Serverless technologies, as well as several words regarding usage and implementations.	Outsourcing operational activities to serverless/cloud providers

12	become, major, continue, increase, becomes, market, power, internet, human, industry, finally, global, begin, consumer, die, online, gain, war, effect, niche, progress, medium, remain, increasingly, growth, attack, law, news, ml, significantly	Clearly not a tech-related topic. Seems to be related to global markets and behavior of humans.	N/A
13	server, instance, ec, cluster, setup, load_balancer, spin, provision, vps, ec_instance, downtime, dns, vpc, redis, load_balance, bare_metal, automatic, take_care, eb, flexible, digitalocean, failover, elastic, auto_scale, bucket, az, ip_address, rout, ridiculous, prem	There is a clear mention of AWS-specific concepts, such as EC2 service instances, Virtual Private Clouds and S3 buckets. EB stands for Elastic Beanstalk, another AWS service. We also see mentions of load balancers and auto-scaling, but also the words "server" and "provisioning", which could indicate the use of automated infrastructure provisioning. Clearly related to NoOps.	Outsourcing operational activities to serverless/cloud providers, Auto-scaling, Automated infrastructure provisioning
14	world, technology, happen, tech, due, lead, space, cause, result, today, likely, replace, current, hit, public, longer, life, force, lose, significant, form, follow, game, home, personal, area, popular, massive, fall, fully	Probably related to the impact of technology	N/A
15	request, region, rule, connection, http, rail, static, snapshot, cloudfront, waf, url, warm, ssl, firewall, droplet, solaris, terminate, mitigate, web_application, hard_drive, aws_account, header, per_day, tcp, dest, origin, data_transfer, cert, additionally, confirm	Probably related to traditional operations activities, but not enough information for further interpretation.	N/A
16	ai, facebook, privacy, vr, energy, end_decade, food, self_drive_car, battery, meat, mass, mainstream, bitcoin, digital, ar, robot, social_medium, social_network, tech_company, drone, society, dominant, medical, dominate, solar, stock, health, healthcare, movie, ubiquitous	Definitely related to application of AI, AR and VR in various fields. Also seems to be related to a social network, and probably their privacy-related issues. Not related to NoOps.	N/A

17	aws, service, cloud, amazon, customer, offer, gcp, account, provider, offering, migrate, cloud_provider, pricing, migration, gce, shut, reliability, cloud_service, outage, beta, gc, aws_azure, manage_service, beat, rackspace, iaa, impression, commitment, compelling, saving	Topic is related to cloud vendors (AWS, Google Cloud Platform, Azure) and their provided services (e.g., Google Compute Engine)	Outsourcing operational activities to serverless/cloud providers
18	test, deploy, production, environment, deployment, task, script, configuration, debug, deployed, prod, command, shell, ci, ssh, execute, deploys, session, puppet, git, directory, environment_variable, locally, terminal, stag, cli, fleet, repository, jenkins, repo	Clearly related to automated deployment, CI tools (Jenkins), Everything-as-Code (Puppet) and Automated Testing	Automated deployment, Continuous Integration tooling, Everything-as-Code, Automated testing
19	php, ruby, erlang, affect, socket, crash, unix, foo, whatsapp, shitty, bar, rollback, transactional, implication, semantics, script_language, side_effect, partly, point_view, elixir, interpreter, cfengine, nasty, function_call, assert, almost_entirely, xmpp, http_jan_get, unsafe, entirely_different	Related to various programming languages	N/A
20	build, docker, image, layer, dependency, package, install, runtime, instal, compile, nginx, docker_image, docker_container, jar, binary, ubuntu, orchestration, final, dockerfile, frontend, plugin, maven, immutable, bundle, artifact, npm, solves, base_image, builder, horizontal_scale	Related to Docker and container orchestration.	Orchestration mechanisms

21	small, hardware, performance, slow, per, fast, second, compute, limit, size, traffic, minute, amount, several, fit, cheap, couple, launch, drop, usage, million, serve, total, io, moment, block, distribute, latency, processing, comparison	Seems to be related to performance metrics, but there is not enough information for interpretation.	N/A
22	need, etc, easy, application, different, example, set, app, solution, instead, able, require, built, machine, single, version, simple, update, handle, multiple, complex, whatever, whole, specific, benefit, complexity, box, maintain, upgrade, thing_like	Not enough information for interpretation	N/A
23	cost, pay, month, free, expensive, hour, cheaper, save, internal, spend, vendor, spending, spent, dedicate, youre, virtual, afford, alone, scalability, mo, full_time, minimum, expertise, headache, fee, spend_time, save_money, good_luck, hobby, round_error	Seems to be related to costs associated to the use of services offered by vendors, but it is unclear whether this discussion is related to the "cost savings" benefit identified in literature or not, due to words such as as "expensive". Still, it can be related to NoOps.	N/A
24	stack, window, boring, sql_server, ha, react, saas, front_end, go_wrong, installation, boring_stack, quirk, inevitably, window_server, weekend, tightly, back_end, web_development, corner, angular, well_document, comfortable, new_shiny, personal_project, every_week, hipster, development_environment, con, antivirus, weekly	Related to the creation of web-related personal projects.	N/A

25	bill, charge, capacity, billing, cap, credit, payment, discount, monthly, reserve, destroy, peak, reserve_instance, xlarge, throttle, flat_rate, threshold, hobbyist, harm, heavy_user, destroyed, hourly, physically, willing_pay, quota, separately, upfront, pricing_model, capacity_need, pay_per	Probably related to the costs incurred for hobby cloud applications. Since we see the expression "reserve instance", this could definitely be related to the usage of cloud services. However, it can not be mapped to a specific finding.	N/A
26	photo, root, download, map, plug, setting, apache, picture, upload, god, tab, wow, damn, mount, annoy, centos, superior, good_reason, readable, nsa, partial, preference, xml, copying, lol, promote, dropbox, folder, env, cached	Not enough information for interpretation. Seems to be a collection of words used in many comments, but without a lot of standalone semantic value.	N/A
27	run, scale, manage, host, node, resource, configure, volume, overhead, reliable, schedule, versus	Related to traditional operations tasks, but it is unclear whether it is actually related in any way to NoOps.	N/A
28	kubernetes, gke, pod, eks, docker_compose, agent, kube, simplify, autopilot, nomad, docker_swarm, swarm, controller, persistent, kubernetes_cluster, learn_curve, joe, stateless, ingres, stateful, service_discovery, web_page, ww, overkill, dead_simple, kubectl, bob, rac, yml, boon	Mainly related to orchestration mechanisms: there are tools such as Kubernetes, Docker Compose and Swarm, as well as Hashicorp Nomad. There are also some vendor-provided orchestration services, such as GKE (Google Kubernetes Engine) or EKS (AWS Elastic Kubernetes Service)	Orchestration mechanisms
29	read, thanks, love, page, link, yeah, interested, reading, author, cool, correct, awesome, please, hear, detail, original, sorry, curious, thank, ago, blog, feedback, hey, blog_post, glad, recommendation, hi, yep, okay, funny	Not enough information for interpretation, but clearly not related to technology	N/A

30	database, mysql, db, postgres, rds, backup, default, postgresql, transaction, master, replication, store_procedure, tune, restore, hint, replicate, writes, mariadb, dynamo, aurora, buffer, take_minute, relational_database, rdbms, replica, filesystem, mongodb, checksum, ddl, mongo	Database-related discussion, not related to NoOps	N/A
31	live, track, rent, train, remote, living, salary, room, safe, air, remotely, adult, income, wage, everyday, premium, signal, sf, isp, grown, survive, red, regional, bay_area, studio, tower, san_francisco, mechanical, equipment, wifi	Related to the life in the Bay Area.	N/A
32	software, development, everyone, term, automate, focus, automation, role, programmer, engineering, skill, culture, fire, position, blame, deep, communication, theory, therefore, fight, wall, meaning, software_development, software_engineer, embed, direction, board, productive, responsibility, communicate	Clearly related to NoOps. There are words such as automation, communication, culture and responsibility. However, it can not be mapped to specific codes, due to the lack of information.	N/A
33	network, networking, mistake, ip, cross, rack, wordpress, proprietary, datacenter, scalable, lazy, gear, vxlan, dc, billion_dollar, control_plane, aid, proven, end_day, high_availability, good_idea, duplicate, army, datacenters, vpn, slice, small_startup, cisco, mpls, blogging	Not enough information for interpretation	N/A

34	container, linux, vm, vms, kernel, virtualization, isolation, lxc, share_host, os, virtual_machine, xen, exploit, cgroups, mainframe, guest, containerize, vmware, contention, hypervisor, kvm, namespaces, page_cache, linux_kernel, cgroup, parity, bsd, isolate, rush, allocation	Seems to be related to virtualization solutions. Related to traditional operational tasks, but not NoOps	N/A
35	lambda, log, api_gateway, aws_lambda, lambda_function, mb, cold_start, dynamodb, execution, endpoint, cloudwatch, nodejs, alert, serverless_framework, side_project, take_second, zappa, resize, invocation, claudia, cloudwatch_log, cold_start_time, async, cron, response_time, gateway, azure_function, boto, web_interface, apex	The discussion seems to be related to serverless-related solutions, especially Function-as-a-Service offerings from various companies (e.g., AWS Lambda, Azure Functions). AWS API Gateway is also mentioned, since it is often used together with Lambda. We also see services such as DynamoDB, which is the fully managed NoSQL database solution offered by AWS. AWS Cloudwatch is also often used together with Lambda for logging purposes. This topic clearly indicates a relation to NoOps, especially the outsourcing of operations to cloud vendors.	Outsourcing operational activities to serverless/cloud providers
36	fix, switch, bug, port, document, critical, reply, discussion, career, immediately, frustrate, contain, advice, tweet, expectation, login, lesson, lie, employ, lawyer, warn, advocate, channel, surprisingly, broke, month_ago, incompetent, snowden, manner, dell	Not enough information for interpretation	N/A
37	store, key, secret, password, ansible, vault, credential, encrypt, encryption, iam, token, hash, auth, hashicorp, audit, km, temporary, username, submit, playbook, metadata, decrypt, tunnel, hashicorp_vault, secret_management, regulatory, user_id, iam_role, truncate, env_var	Related to Infrastructure-as-Code (Ansible), as well as the management of secrets (store, IAM, encryption, Hashicorp Vault, credential, secret_management)	Everything-as-Code

38	china, country, government, rise, political, climate_change, economy, economic, eu, population, europe, recession, tax, least_one, decline, social, usa, regulation, india, mar, american, collapse, chinese, africa, currency, nation, housing, debt, politics, united_state	Geopolitical discussion related to climate change and economy.	N/A
39	generate, id, yaml, random, output, string, declarative, dsl, general_purpose, counter, unit_test, uuids, primary_key, chart, generator, uuid, helm, prevents, haskell, construct, sequence, simultaneously, binding, markup, dhall, char, roll_back, per_second, integer, collision	Generation of random strings? Not enough information for interpretation.	N/A
40	write, design, requirement, define, concept, text, definition, describe, reference, input, magic, context, html, constraint, frequently, perfectly, btw, tag, truth, ideal, hence, new_feature, described, assumption, oppose, little_bit, abstract_away, description, end_user, odd	Not enough information for interpretation.	N/A
41	computer, rewrite, flow, convince, wheel, operator, whose, student, dream, yr, cobol, character, speech, surprising, arrive, heart, worried, pair, radar, concrete, intelligent, stress, theme, designer, recall, earlier, server_side, tube, mechanic, obtain	Not enough information for interpretation.	N/A
42	device, apple, phone, information, browser, desktop, mobile, mac, android, bank, laptop, count, conference, pc, location, render, listen, cable, card, english, bias, abandon, iphone, hurt, visit, keyboard, usb, london, fa, macbook	Seems to be related to electronic devices	N/A

43	price, oil, double, supply, raise, estimate, profit, capital, commodity, asset, expense, adjust, margin, massively, short_term, study, mile, floor, elasticity, lower, dram, knock, barrel, aggressive, last_week, reduces, ceiling, inelastic, reduce_cost, large_organization	The fluctuations in the price of oil	N/A
44	data, query, table, sql, redshift, bigquery, record, join, index, column, select, insert, row, schema, partition, snowflake, bq, dataset, warehouse, scan, reporting, sample, etl, spectrum, historical, concurrency, along_line, dual, extensive, pg	Data warehousing solutions and their usage	N/A
45	decade, prediction, car, city, drive, driver, bet, road, predict, hype, vehicle, electric_car, tesla, next_year, innovation, self_drive, electric, agi, wasm, condition, anti, autonomous, adoption, next_decade, earth, widely, travel, truck, station, child	Electric cars and autonomous vehicles	N/A
46	uk, co, russia, emission, germany, union, brexit, per_caput, recover, retire, flood, economically, australia, st, soft, boy, euro, france, carbon_emission, paris, putin, rd_world_country, singapore, world_country, cost_center, china_india, bankruptcy, st_world, iran, italy	Climate change-related geopolitical discussion	N/A
47	devops, team, ops, operation, engineer, hire, devs, sysadmin, organization, noops, sysadmins, title, staff, admin, ops_team, devops_engineer, infra, agile, department, empathy, silo, ops_people, devops_team, small_company, dev_ops, specialist, principle, qa, mindset, dev_team	Interesting enough, the concepts DevOps and NoOps are mentioned, as well as concepts related to operations teams. The word silo could refer to the traditional silo between development and operations ("devs" and "ops").	Removing the silo between Dev and ITOps teams

48	google, google_cloud, twitter, spotify, partner, music, big_data, acquisition, gmail, cloud_platform, customer_support, disclosure_work, data_studio, bi, announcement, loud, bigtable, looker, reputation, sm, paid_support, notify, engagement, partnership, hangout, play_store, web_ui, dataproc, case_study, live_migration	Clearly related to the services offered by Google Cloud, and how Spotify (which is a client of Google Cloud Platform) uses these services.	Outsourcing operational activities to serverless/cloud providers
49	storage, memory, load, workload, disk, cpu, ram, cache, gb, hundred, benchmark, tb, concurrent, silly, hadoop, spark, ten, ssd, allocate, swap, single_machine, throughput, fit_ram, cost_per, utilization, max, min, gb_ram, terabyte, fetch	Resources required for Big Data processing	N/A
50	java, oracle, complain, jvm, teach, jdk, clojure, compatibility, scala, compatible, linq, perl, optional, backwards, complainer, old_version, eol, observation, maintainer, yay, unsupported, statically_typed, low_code_solution, backward, original_comment, moot, mature_enough, openjdk, java_jvm, mandate	Discussion related to programming languages based on the Java platform	N/A
51	linode, bandwidth, digital_ocean, data_center, per_month, cdn, hetzner, free_tier, cloudflare, dedicate_server, much_cheaper, metal, segment, physical_hardware, api_call, colo, softlayer, lightsail, low_price, keep_mind, egress, hr, cdns, across_multiple, big_advantage, per_hour, early_day, emr, vultr, dedicate_hardware	Discussion related to vendors that offer dedicated servers.	N/A

52	terraform, queue, cloudformation, fargate, template, sqs, elb, convox, cf, ami, kinesis, autoscaling, consul, use_terraform, security_group, cloudfoundry, aws_fargate, hcl, autoscaling_group, kinesis_stream, cloudformation_template, reusable, trigger_lambda, codebuild, cf_template, asg, logstash, sn_sqs, scale_horizontally, fud	The words "Terraform", "HCL" (Hashicorp Configuration Language) and "CloudFormation" suggest the use of Infrastructure-as-Code. There are also some PaaS solutions mentioned, such as CloudFoundry and Convox. AWS Fargate is a Serverless compute engine, which completely abstracts away infrastructure details. AWS Kinesis enables distributed data processing of streaming data. SQS is another AWS service which provides message queueing service. Finally, there are some mentions of AWS Lambda, as well as auto-scaling. Clearly related to NoOps.	Outsourcing operational activities to serverless/cloud providers, Everything-as-Code, Auto-scaling
53	net, net_core, cross_platform, asp_net_core, ii, mono, dotnet, restriction, mvc, wpf, asp_net, ef, electron, xamarin, vscode, rider, ef_core, injection, versioning, cough, net_framework, vim, maintenance_mode, store_procs, netfx, autocomplete, winforms, razor, erp, whitelist	Seems to be related to the dotnet ecosystem	N/A
54	name, hack, email, com, send, contact, format, represent, resume, info, recruiter, year_old, ex, credit_card, register, reminds, grade, stack_overflow, math, survey, headline, central, flip, six, press, gbps, length, weak, www, pdf	Recruitment-related topic	N/A
55	app_engine, gae, appengine, datastore, manage_vms, internally, google_cloud_platform, google_app_engine, compute_engine, django, disclaimer_work, cloud_sql, manage_vm, snapchat, google_compute_engine, container_engine, emulate, deprecate, runtimes, cloud_storage, ga, par, pubsub, private_network, task_queue, rabbitmq, flexible_environment, cloudsql, rel, nofollow	Related to Google Cloud services, such as AppEngine, PubSub and Compute Engine.	Outsourcing operational activities to serverless/cloud providers

Table G.1: Interpretation of all topics resulted from the topic modeling

Appendix H

Seminar report

NoOps Systematic Literature Study

Student
Dragos Mihai Serban

Supervisors
Alexander Serebrenik
Damian A. Tamburri

Eindhoven, January 31, 2021

1 Introduction and problem description

Over the past few decades, the ways people build and operate software have changed dramatically. While linear models, such as the V-Model or the Waterfall approach, were very popular in the early days of software engineering, they have been widely criticized for multiple reasons, including the high level of rigidity, lack of iterations and, most importantly, the lack of collaboration between teams. In the Waterfall approach, for example, it was common to have a different team for each stage of the process, from requirements engineering to operation and maintenance. The high rigidity of the model did not allow overlapping responsibilities between different stages. Hence, the end results of each stage were directly passed to another team, which was responsible for the next stage of the process [1]. In turn, this severe separation between teams led to improper communication, which resulted in high costs for addressing unexpected changes or issues [2].

The disadvantages of these traditional models led to the creation of the Agile methodology, which promised to bring value to the customers in a quicker way, with improved collaboration and cross-functional teams. This methodology revolutionized the software development world by proposing a number of principles, such as frequent software delivery, shorter development timelines, continuous delivery, and welcoming new requirements at any point in time [3]. Although the Agile methodology was overall very successful, it could not be deemed as perfect. One of the reasons is that Agile was oriented more towards software development, which caused a lack of involvement from an operations perspective. The software that was being produced could not be delivered as quickly [4].

This lack of collaboration between software development and operations teams served as the starting point for DevOps. At its core, DevOps is a set of practices that aims at bringing the development and operations teams together, such that they can build, test and release software quicker [5]. The DevOps methodology integrates both technical practices, such as Continuous Integration, Continuous Delivery, Infrastructure as Code, Monitoring and Logging [6], but also cultural practices, such as open collaboration, responsibility alignment and continuous learning. However, as pointed out by Mike Gualtieri in a controversial article, a major problem of DevOps is the assumption that developers want to become involved in operations or even sit down with operations people [7]. This assumption led to the conception of NoOps (short for No Operations).

NoOps, which is a term coined in 2011 [7], promises to let developers focus solely on development activities, thus removing the need to perform any operational activities. However, NoOps is still a vague concept, lacking a formal definition and a definitive set of practices. In addition, at the moment of writing there are no peer-reviewed works on the concept of NoOps.

This report is an exploratory study, aiming at providing a preliminary overview of the NoOps concept. In particular, this study has three main goals: exploring the different perspectives of practitioners over the definition of NoOps, creating an overview of the characteristics that can be associated to NoOps, as well as providing an outline of the problems and challenges that practitioners associate with NoOps. In order to achieve these goals, a systematic literature mapping study based on gray literature is conducted. Gray literature is defined as follows: literature that ”is produced on all levels of government, academics, business and industry in print and electronic formats, but which is not controlled by commercial publishers, i.e., where publishing is not the primary activity of the producing body” [8]. The reason for focusing on gray literature is the fact that there is absolutely no peer-reviewed content on the concept of NoOps, as previously mentioned. The results of this study are relevant for researchers, since a number of challenges are introduced, but also for practitioners.

The rest of the report is structured as follows. Section 2 presents some studies that are related to the work presented in the current report. Section 3 describes the methodology that was used while conducting the study, while section 4 illustrates the obtained results. These results are later discussed in section 5. Finally, section 6 outlines several conclusions.

2 Related work

Considering the lack of peer-reviewed content related to NoOps, this section will provide a brief overview of the use of gray literature in software engineering research.

Garousi et al. [9] proposed a set of guidelines for including gray literature and conducting systematic literature reviews in software engineering. These guidelines were created based on existing systematic literature review guidelines in software engineering that focus solely on peer-reviewed literature. The motivation of the authors for including gray literature in systematic literature studies is the assumption that practitioner literature could improve the relevance of software engineering research. These guidelines also face some limitations, such as lack of empirical evaluation, as well as subjectivity, since they are largely based on the personal experience of the authors.

In another paper, Garousi et al. [10] highlight the main challenges associated with using gray literature in software engineering research, namely the lack of formal definitions and models of gray literature materials, the large quantity and variability of material, as well as the lack of proper quality checklists.

Yasin et al. [11] propose a categorization scheme for gray literature in software engineering, as well as a quality checklist that could be used for filtering purposes in systematic literature reviews. In addition, the authors show that Google Scholar is a reliable tool for retrieving peer-reviewed primary studies.

3 Methodology

This section introduces the methodology behind the conducted systematic literature mapping study. In order to ensure the validity and reliability of the presented approach, the guidelines of Petersen et al. [12] for conducting systematic literature mapping studies have been used. However, due to the lack of peer-reviewed material to be used as primary studies, the guidelines of Garousi et al. [9] for including gray literature in systematic literature studies have been used as well.

The overall process used to perform this systematic literature mapping study, after the definition of the research questions, is depicted in Figure 1. In short, the first step of the process is performing a search for studies on multiple search engines. These initial search results are then filtered based on a set of inclusion and exclusion criteria. Then, iterative forward snowballing is applied to the remaining results, meaning that all resources that are referenced in this initial set of studies are added to a pool of potential candidates for primary studies. This pool of potential candidates is also filtered based on the same set of inclusion and exclusion criteria. After snowballing does not provide any new results, all the remaining studies are subject to a quality control process. Afterwards, data extraction and analysis are performed on the resulted set of primary studies. In order to control for subjectivity, inter-rater reliability measurements are conducted at multiple stages: after the application of inclusion and exclusion criteria, after quality control and during the analysis phase.

The rest of this section is structured as follows. Section 3.1 introduces the research questions and defines the scope of this study. Section 3.2 describes the literature search process. Section 3.3 highlights the factors that have been used for sample selection and for controlling the sample quality. Finally, section 3.4 introduces the process used to extract, synthesize and analyze the data.

3.1 Research questions

The main goal of this study is to provide a preliminary overview of the concept of NoOps from the perspective of practitioners. In particular, the study aims at eliciting the definition of NoOps, the advantages of the approach, the main characteristics and practices associated with the concept, as well as the challenges, problems and fallacies related to NoOps. That being said, the following research questions are formulated:

RQ1. How do practitioners describe NoOps? - This research question aims at obtaining an overview of the different perspectives of practitioners regarding the definition of NoOps, as well as the expected and reported advantages associated to the adoption of NoOps.
RQ2. What are the characteristics of NoOps? - The goal of this research question is to elicit the main technical, social and cultural characteristics of NoOps.
RQ3. What are the challenges related to NoOps that are reported by practitioners? - This question aims at providing a high-level overview of the main challenges, problems and fallacies related to NoOps.

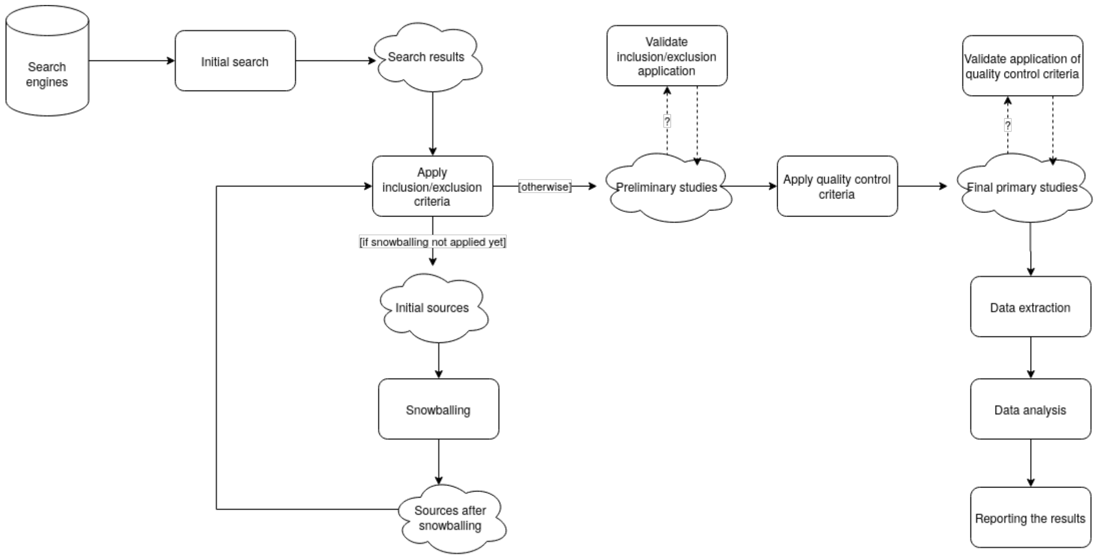


Figure 1: The general workflow for conducting the systematic literature mapping, after the definition of the research questions

3.2 Literature Search process

Based on the recommendations of Garousi et al. [9], gray literature can be found using mainstream search engines, such as Google, Bing or Yahoo. For the scope of this study, the decision was to use Google, Bing and Google Scholar as primary search engines. Google and Bing have been chosen due to their high popularity and usage in the English-speaking world. Google Scholar has been chosen in order to find gray literature related to NoOps that might have been referenced in peer-reviewed content, according to the guidelines of Yasin et al. [11]. Using the guidelines of Kitchenham et al. [13], a number of search strings have been created based on the main keywords of each research question, as listed below:

("NoOps" OR "No-Ops" OR "No Ops") AND ("definition" OR "characteristics" OR "practices" OR "culture" OR "challenges")

The search strings above have been used identically on all three search engines. In addition, the search parameter "Only English results" has been used on all three search engines.

For each search string, the first 20 pages of results for each search engine have been considered. Since all the three search engines list the results in the order of relevance, the assumption was that all results returned after the first 20 pages would be very irrelevant.

After obtaining all search results, the duplicates have been merged based on the URL.

3.3 Sample selection and quality control

Given the large number of irrelevant results provided by the initial literature search process, a manual screening of the results has been performed. This manual screening was based on a number of inclusion and exclusion criteria that have been defined, as presented in Table 1.

The filtering process started with the application of the inclusion criteria. These criteria have been applied sequentially, one-by-one, starting from i_1. The inclusion criteria have been applied based on the information presented in the title and the meta-description of the search results in the case of $i_1 - i_3$, but also on the entire resource content in the case of i_4. It is important to note that although i_1 to i_3 are objective criteria, i_4 is a subjective criterion, since relevance could be assessed differently by other researchers. To increase the reliability of our results, we have decided to measure inter-rater reliability with the author of this report and

Inclusion	i_1. Resource is in English i_2. Explicit discussion of NoOps as set of practices within software engineering i_3. Located on the first 20 pages of search results i_4. The content of the resource is related to at least one of the research questions
Exclusion	e_1. Resources that are behind a paywall e_2. Resources that are completely product or service advertisements, without providing any additional insights related to NoOps e_3. Resources without a clear publication date e_4. Resources without explicit authorship e_5. Resources that are not available (do not exist) e_6. Resources that are duplicates

Table 1: Inclusion and exclusion criteria

the two supervisors. To evaluate the inter-rater reliability, we have agreed to compute the Krippendorff's alpha coefficient [14], which is used to measure the agreement between multiple lists of codes applied by multiple raters. We have decided to use this coefficient specifically because it could be applied for more than two raters, in contrast to other coefficients, such as Cohen's kappa [15].

After the application of the inclusion criteria, the filtering process continued with the application of the exclusion criteria. These criteria have also been applied sequentially, one-by-one, starting from e_1. Please note that criterion e_3 has been considered necessary because it would enable the mapping of results and resources to a specific timeline. On the other hand, e_4 has been considered because the author names are required in the quality control stage, in order to establish their level of expertise.

Finally, the resulted list of primary studies has been subjected to quality control, as recommended by Garousi et al. [9]. The reason why additional quality control is recommended in our case, in contrast to traditional systematic literature studies, is that gray literature is not peer-reviewed, therefore the provided evidence can not be fully trusted. In opposition to the approach of Garousi et al., our quality control does not aim at eliminating resources based on a quality threshold. In fact, the goal of our quality control is to sort the resources in descending order, based on the quality score. This sorted list is used to define the order of analysis for the resources. The assumption, in this case, is that this form of purposive sampling would aid the process of reaching theoretical saturation in the analysis phase, since it is expected that resources that have a higher quality score would also provide more in-depth results, resulting in a faster reach of theoretical saturation.

The quality control has been applied based on a variant of the quality checklist proposed by Garousi et al. [9]. To be more specific, the authors proposed a quality checklist with 20 different items, aiming at assessing various quality criteria. From this checklist, we have selected 6 items that could be applied for this study and adapted them to the current context. In particular, in the quality control phase the goal is to answer the following questions:

Q_1. Is the author of the resource a practitioner?

Q_2. Does the publisher review content before publication?

Q_3. Is the author associated to a software or cloud-related company or organization?

Q_4. Does the author generally publish works in the field of DevOps, Serverless, NoOps, ITOps, or Software Engineering in general?

Q_5. Does the resource provide any form of evidence for the claims?

Q_6. Is the source type reputable?

In Q_1, by practitioner we refer to any job related to software engineering, DevOps, or cloud computing, even if it is a managerial or a technical job. The goal of this question is to determine whether the public discourse reflects a real practitioner perspective, rather than the perspective of non-technical individuals,

4

such as journalists or content writers. In Q_2, the goal is finding out whether the publishing outlet performs any sort of content reviewing before publication. For instance, while magazine articles and whitepapers are most definitely reviewed before publication, this might not be the case for ad-hoc blog posts. Q_3 and Q_4 aim at investigating whether the author has any relation to a specific software company and if the author generally publishes other works related to software engineering. Then, Q_5 aims at investigating whether the presented claims are supported by any form of evidence. In particular, the focus of this question is on the presence of references or any type of evidence described in the hierarchy of Alves et al. [16]. Finally, Q_6 is assessed based on the scoring guideline of Garousi et al. [9], which implies that the resources which have a high outlet control and credibility (e.g. books and whitepapers) are assigned a higher score than resources that have a moderate or low outlet control and credibility (e.g. blog articles or podcasts).

For all these questions, a score of either 0 or 1 is assigned (1 if the answer to a question is yes, 0 otherwise). In the case of Q_6, the scores can be either 0, 0.5 or 1, based on the scoring system suggested by Garousi et al. [9]. Then, an average for all these scores have been computed per resource, obtaining a normalized quality score. Finally, the entire list of resources is sorted in a descending order based on this normalized quality score. Please note that the questions Q_1 to Q_4 have been answered by checking the online presence of the author via search engines and social media platforms.

3.4 Data extraction & analysis

In the data extraction phase, the list of resources is organized into smaller batches of 10 resources, based on their quality score. Then, for each resource in a batch, the content is extracted per paragraph into an Excel worksheet. Moreover, in order to record whose voice is reflected, the author's job title is extracted as well. In the case of video and audio resources, the content has been transcribed based on the "chunk transcription" method presented in [17].

The analysis process involves open coding and it has been performed according to the guidelines of Wagner and Fernandez [18]. Open coding is a qualitative analysis method which involves assigning labels (also called codes) to small fragments of text and categorizing them into a hierarchy. Open coding is performed per batch and is repeated until a state of theoretical saturation is reached. In other words, the open coding process is stopped only when no more codes can be created.

In this study, open coding is performed per batch of 10 resources and it involves 2 iterations per batch. The first iteration has the purpose of identifying the paragraphs of a resource that could be relevant for a specific question. In this first iteration, for each paragraph of a resource in the batch, we either assign a "macro-code" or we mark the paragraph as "Not Applicable". The macro-code represents a higher-level category that could encompass the content of a paragraph. For instance, if a paragraph would provide a definition of NoOps, then this paragraph would be assigned the code "RQ1. NoOps definition", since the paragraph is relevant for the first research question and it contains a definition for the concept. On the other hand, the second iteration has the goal of assigning "micro-codes" for all the paragraphs that have been identified as relevant in the previous iteration. In other words, this iteration involves assigning some very specific labels that would precisely represent the content of the paragraph. For instance, when a paragraph discussed about automated monitoring as a technical characteristic of NoOps, then the label "Automated Monitoring" has been assigned.

After each batch has been analyzed, the micro-codes have been categorized into high-level cohesive groups using an approach called card sorting. Card sorting is a method that is widely used in qualitative analysis for assigning higher-level categories to fragments of text. This method involves placing the fragments of text (micro-codes in the current context) onto individual pieces of paper (so-called cards), that are then iteratively sorted into groups. We have opted for open card sorting, meaning that the groups emerged naturally during the sorting (i.e. the groups were not predefined). In this process, the guidelines of Zimmerman have been used [19].

The categories that resulted from card sorting have later been used to create mind map diagrams. The purpose of the mind map diagrams is to illustrate the taxonomy of concepts.

4 Results

4.1 The search for primary studies

Appendix A illustrates the number of results obtained for each search string per search engine. All the results have been collected in a single day, on December 8, 2020.

Overall, 2182 results have been returned. After merging the search results based on their URL, 971 unique resources remained. These resources have then been subject to the application of inclusion and exclusion criteria. The statistics resulted from the application of inclusion criteria can be found in Table 2. As it can be observed, after the application of all inclusion criteria, only 133 resources remained. After the application of exclusion criteria on these remaining resources, 95 resources remained. On these 95 resources, forward snowballing has been applied in two iterations, revealing 3 additional resources. The final list of 98 primary studies can be found in Appendix B.

Inclusion criterion	Description	# before	# after
i_1	Resource is in English	971	844
i_2	Resource provides an explicit discussion of NoOps as set of practices within software engineering	844	286
i_3	Resource is located on the first 20 pages of results	286	286
i_4	Resource is relevant to at least one of the research questions	286	133

Table 2: The number of resources before and after the application of each inclusion criterion

4.2 Overview of selected primary studies

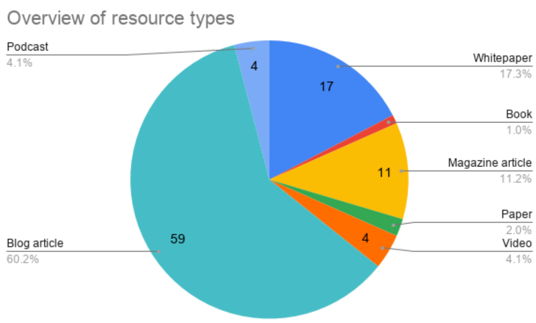
In this section, an overview of the selected primary studies is provided.

Figure 2 illustrates the distribution of the resources based on the publishing year. The first mentions of the term NoOps appeared in 2011. Although in 2013 there have been no resources related to NoOps, we can see that from 2014 the number of resources increased significantly, reaching a peak in 2019, when 26 resources have been published.

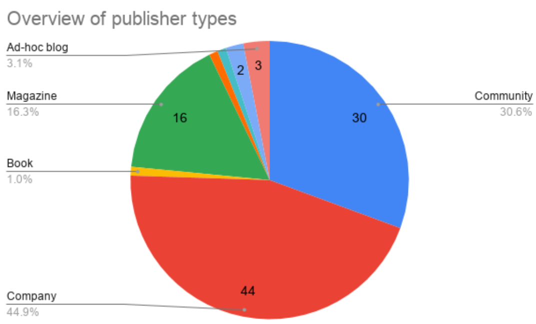
Figure 3b presents an overview of the types of resource publishers. Eight types of publishers have been identified: companies, book publishers, magazines, ad-hoc blogs, communities, independent podcasts, practitioner conferences and scientific journals. While companies seem to have published almost half of all considered resources (e.g. on company websites), approximately a quarter of all resources have been published in communities (platforms that allow anyone to contribute). The publication trends per type of publisher are presented in Figure 4b.

Regarding the types of resources, as shown in Figure 3a, the following types have been identified: blog articles, podcasts, whitepapers, books, magazine articles, papers and videos. While blog articles seem to be the majority, there is also a considerable number of whitepapers and magazine articles. The publication trends per type of resource are presented in Figure 4a.

Figure 2: The number of resources published per year



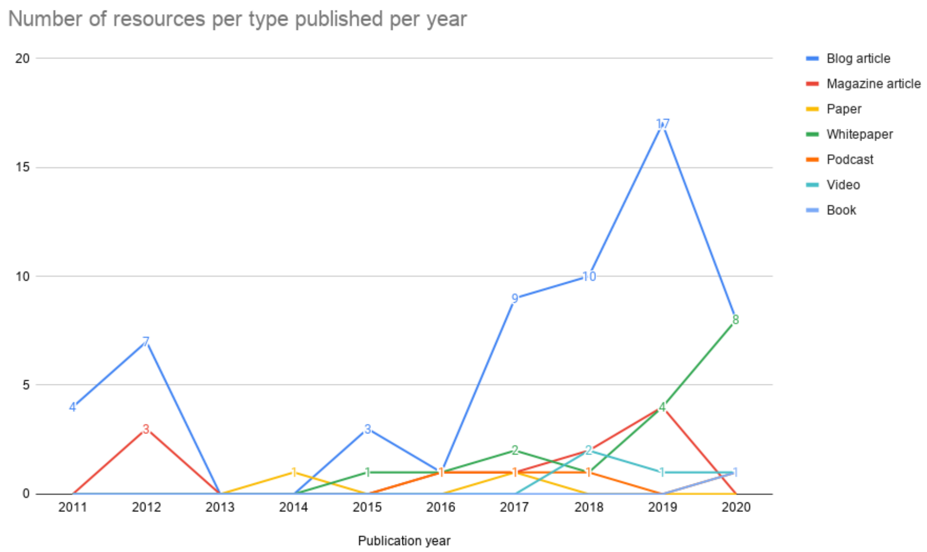
(a) Overview of resource types



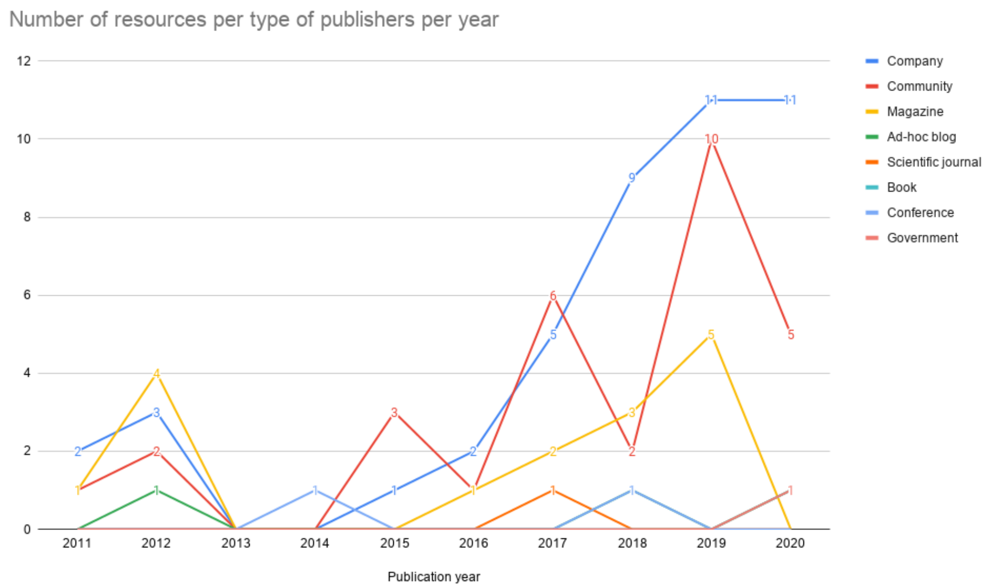
(b) Overview of publisher types

Figure 3: An overview of the resource and publisher types

7



(a) The number of resources published per year (by type)



(b) The number of resources of resources published per year (by type of publisher)

Figure 4: An overview of resource publication trends

4.3 Analysis results

In this section, an overview of the results obtained during the analysis phase will be given. The analysis is limited to the first 30 resources from Appendix B. In addition, the mapping of resources to findings can be

8

found in Appendix D.

4.3.1 RQ1. How do practitioners describe NoOps?

Overall, for the first research question, two main categories of codes have been identified in the studied literature: NoOps definition and NoOps advantages.

The different perspectives regarding the definition of NoOps can be observed in Table 3. The most frequent perspective is that NoOps is a set of practices that is complementary to DevOps. Another popular perspective is that NoOps is a concept which implies that developers do not have to interact with the operations team at all. Another perspective describes NoOps as a mindset rather than as a practice, meaning that no traditional operational tasks are necessary to deploy and operate software. Other definitions imply that the IT environment is highly automated (and potentially abstracted away from the underlying infrastructure), such that there is either a reduced necessity or no necessity at all for operations.

Figure 5 illustrates a mind map diagram highlighting the main perspectives, as well as the relationships between some of them them. This mind map diagram represents the outcome of the card sorting process described in the previous section.

Code	**Frequency**
NoOps implies that an IT environment is very automated, such that there is no necessity for a dedicated team to manage operations in-house	1
NoOps is the goal of completely automating the deployment, monitoring, management of applications and infrastructure	1
NoOps implies that an IT environment is very automated and abstracted from the underlying infrastructure, such that there is a reduced necessity for a dedicated team to manage operations in-house	2
NoOps implies that an IT environment is very automated, such that there is a reduced necessity for a dedicated team to manage operations in-house	4
NoOps implies that developers do not have to interact with the operations team	4
NoOps implies that an IT environment is very automated and abstracted from the underlying infrastructure, such that there no necessity for a dedicated team to manage operations in-house	5
NoOps is the mindset that no traditional operational tasks are necessary to deploy and operate software	5
NoOps implies total exclusion of IT Ops	6
NoOps is complementary to DevOps	15

Table 3: The frequency of codes (perspectives) related to the NoOps definition

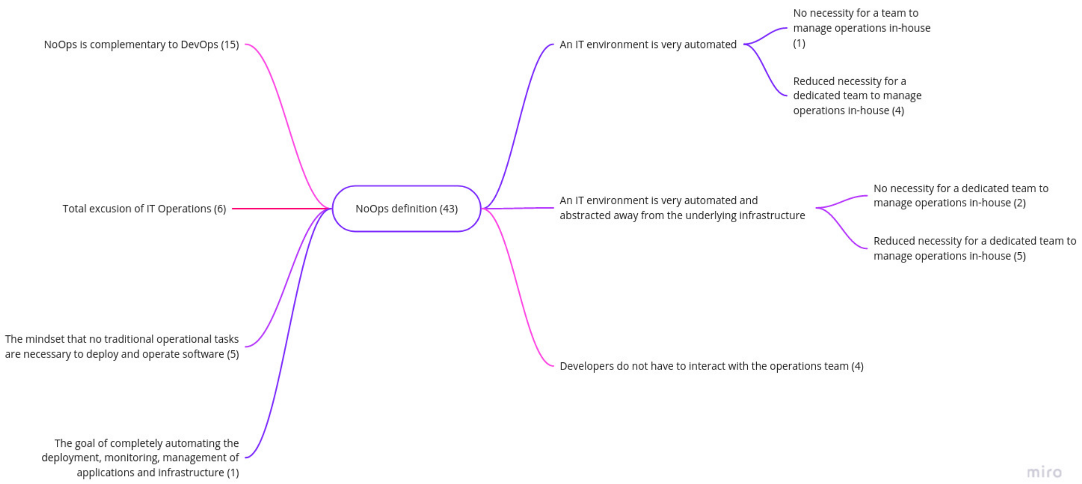


Figure 5: Mind map diagram highlighting the main dimensions of the NoOps definition. The numbers in brackets represent the frequency of a dimension in the analyzed literature

Table 4 presents the NoOps advantages that have been reported in the analyzed literature. These advantages have been grouped into 4 higher-level categories through card sorting, the result being illustrated in Figure 6. As it can be observed, the categories of advantages are related to: operational processes, organization and business, social and technical (development). Advantages regarding the operational processes are the most common, followed by social, organizational and business and, lastly, technical (development) advantages.

Code	**Frequency**
Provides scalability	1
Facilitates testing automation	1
Lowers the risk for cyberattacks	1
Reduces necessity for scaling ITOps team size	1
Improves collaboration	1
Enables self-service	1
Increases accountability	1
Build in Governance, Observability, Quality and Compliance	1
Enhances operations process with further intelligence, reducing the need for incident responses	2
Facilitates quality engineering	2
Enhances reliability	2
Incentivizes developers to write better quality code	2
Minimizes the risk of human error	3
More time and opportunities for learning and growth for ITOps team	4
Cost savings for infrastructure and application operations	5
Additional operational efficiency for infrastructure and application operations	5
Eliminates friction with the infrastructure operations (especially in the release phase), such that developers can create value by not having to interoperate with operations teams	7

Table 4: The frequency of codes related to the NoOps advantages

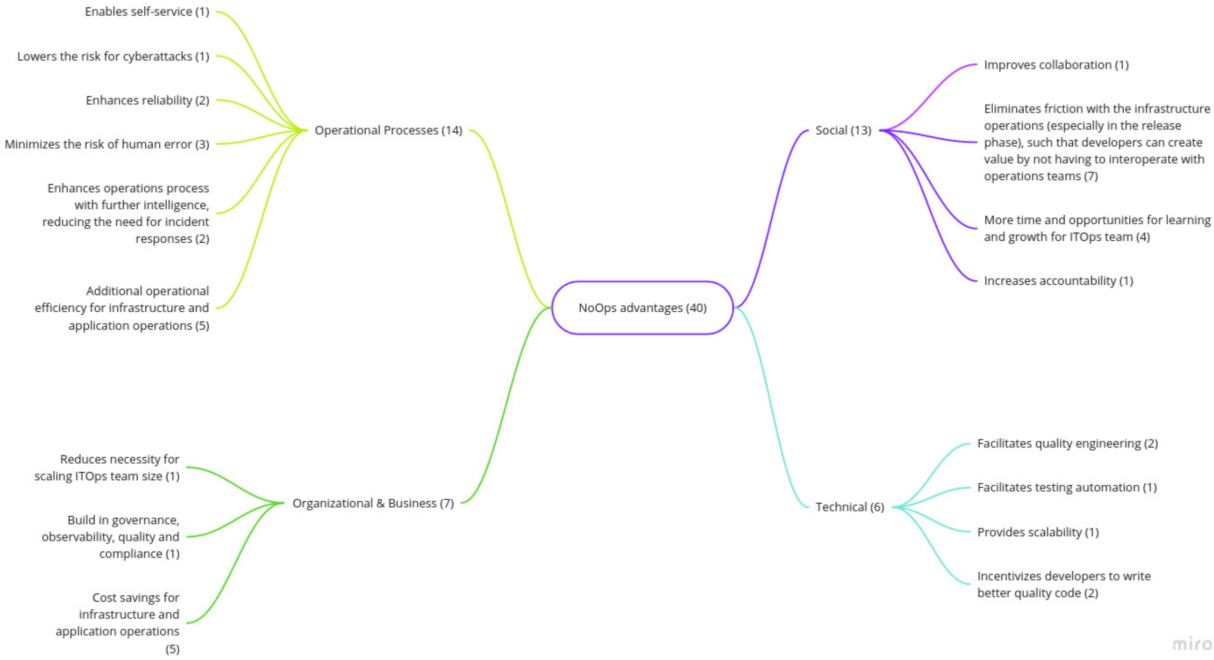


Figure 6: Mind map diagram highlighting the NoOps advantages and the identified categories. The numbers in brackets represent the number of occurrences of a specific advantage in literature.

4.3.2 RQ2. What are the characteristics of NoOps?

Overall, for the second research question, we have identified two types of characteristics: technical and cultural. The main findings related to technical characteristics can be found in Table 7, while the findings regarding cultural characteristics can be found in Table 8.

When it comes to technical characteristics, these have been categorized into 4 main categories: automation aspects, architectural, tooling and practices. The practices category has been further divided into general and NoOps-specific practices, as shown in Figure 7. The most common technical characteristics discussed in literature are related to (NoOps-specific) practices, followed by automation aspects, tooling and architectural characteristics.

11

Code	**Frequency**
Cost-benefit analysis for deciding which aspects to automate	1
Quality engineering	1
Virtual assistants	1
Support for Site Reliability Engineering	1
Orchestration mechanisms	1
Microservice architecture suitability	1
Chaos engineering	1
Automated security scans	2
Automation of test-release-operate phases	2
Automated testing	2
Automated reporting of critical information	2
Software bots	2
Continuous Delivery	2
Ensuring service observability	2
Continuous Integration tooling	3
Build automation	3
Self-diagnostics	3
Auto-scaling	3
Automated management of applications	3
Automated infrastructure provisioning	4
Implementing feedback cycles	4
Self-service portals for operations	4
AI/ML-based automation of operations	5
Proactive monitoring	5
Self-healing	6
Everything-as-Code	8
Automated deployment	8
Automated monitoring	14
Outsourcing operational activities to serverless/cloud providers	30

Table 5: The frequency of codes related to the NoOps technical characteristics

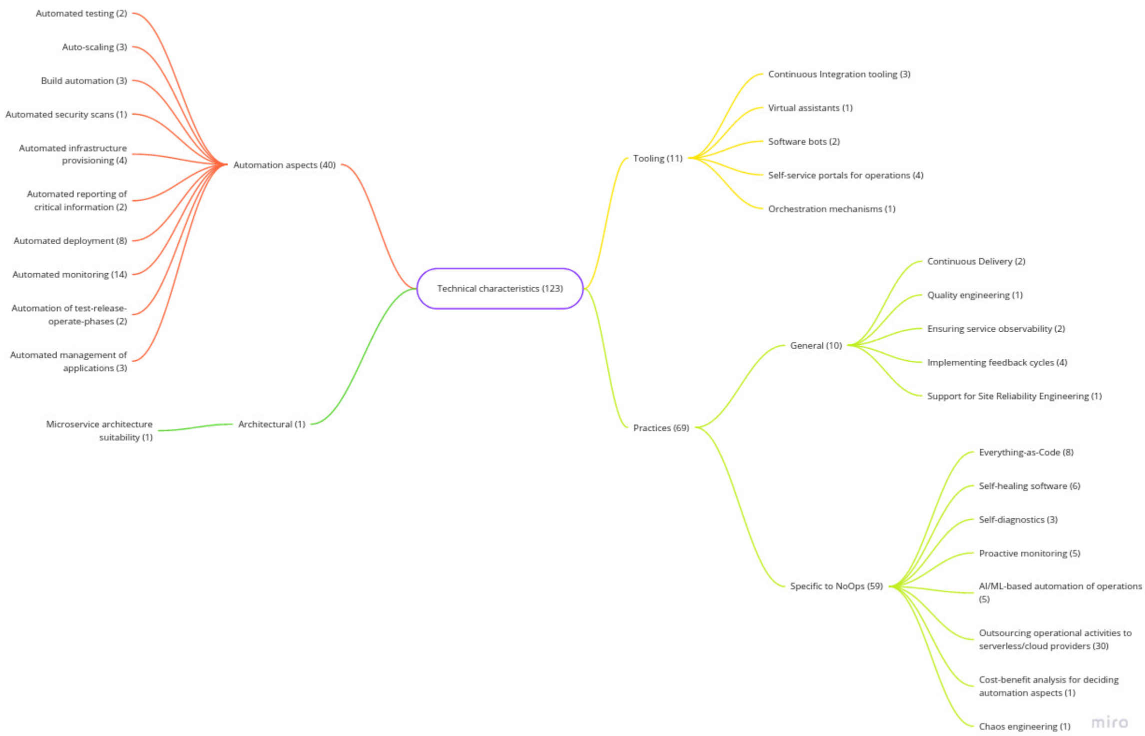


Figure 7: A mind map diagram highlighting the NoOps technical characteristics. The numbers in the brackets represent the frequency of each characteristic in the analyzed literature.

When it comes to cultural characteristics, these have also been divided into four main categories: collaborative, organizational, responsibility shifting and mindset. As it can be noticed in Figure 8, the most frequent characteristics are related to shifting various operational responsibilities towards developers, followed by collaborative characteristics, organizational and mindset.

Code	Frequency
Aiming for process reproducibility	1
Proactivity in engineering the variability of operations	1
Removing the walls between IT and business	1
Shifting developers' thoughts towards taking ownership of the platforms	1
Shifting responsibility for operations from ITOps teams to Software Architects	1
Shifting monitoring responsibility from the ITOps teams to the development teams	2
Shifting security operations from production stage to development stage	2
Training teams on new automation tools	2
Transfer of skills	3
Shifting quality assurance from production stage to development stage	3
Management involvement and acceptance	3
Removing the silo between Dev and ITOps teams	7
Responsibility for deployment and operation is shifted to software engineering teams	8

Table 6: The frequency of codes related to the NoOps cultural characteristics

13

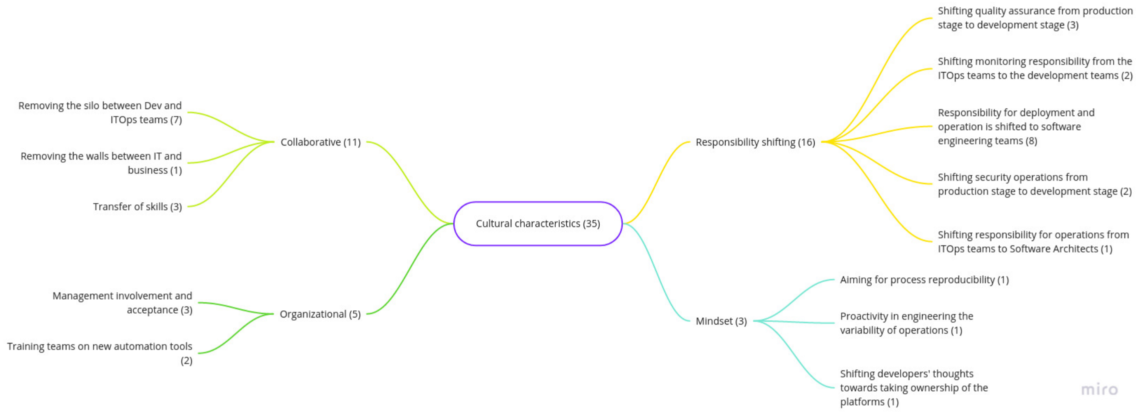


Figure 8: A mindmap highlighting the NoOps cultural characteristics. The numbers in the brackets represent the frequency of each characteristic in the analyzed literature.

4.3.3 RQ3. What are the challenges related to NoOps that are reported by practitioners?

For the third research question, five types of challenges have been identified: social, technical, cultural, environmental and general challenges. The findings for each category of challenges can be observed in tables 7, 8, 9, 10, and respectively Table 11. A full overview of the challenges associated to NoOps can be found in Figure 9.

As it can be observed in Figure 9, most of the reported challenges are of a technical nature, followed by challenges of a social nature. When it comes to cultural, environmental and general challenges, only one challenge per category has been identified.

Code	Frequency
NoOps can not be based on traditional ways of working together	1
Reported hesitation to understanding the business in the initial phases of adoption	2
Reported discomfort in the initial phases of adoption	3

Table 7: The frequency of codes related to the NoOps social challenges

Code	Frequency
Management of functions when using FaaS solutions	1
Poor NoOps implementation quality could lead to cyber risk at scale	2
Vendor lock-in	2
Outsourcing activities to third-parties could lead to technical issues and limitations	3
Adoption may not be suitable or requires re-engineering of legacy systems	15

Table 8: The frequency of codes related to the NoOps technical challenges

Code	Frequency
NoOps requires a mindset shift for developers in order to ensure that the developed artifact behaves the same in both local and production environments	2

Table 9: The frequency of codes related to the NoOps cultural challenges

Code	Frequency
Regulatory compliance, security and other cross-cutting concerns need to be addressed company-wide	3

Table 10: The frequency of codes related to the NoOps environmental challenges

Code	Frequency
Adopting NoOps could lead to less involvement from development in operations, which completely contradicts DevOps practices	2

Table 11: The frequency of codes related to the NoOps general challenges

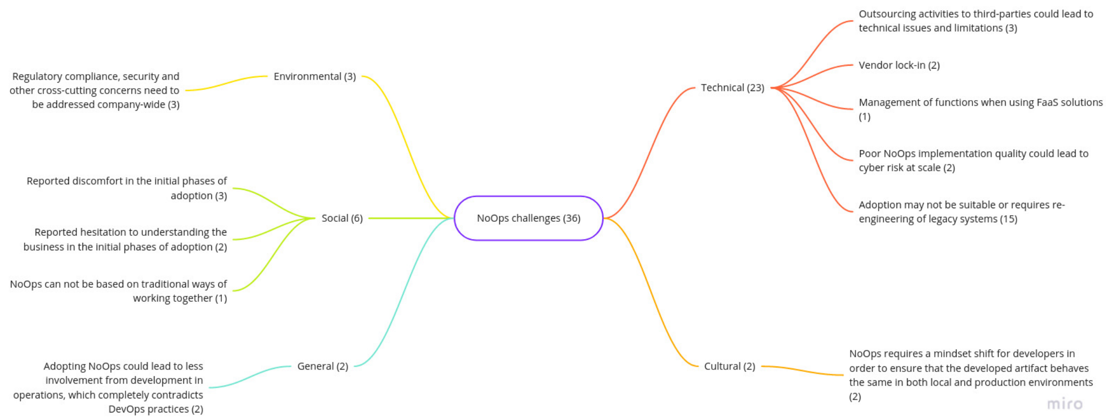


Figure 9: A mind map diagram highlighting the NoOps challenges reported in literature. The numbers in parentheses represent the number of occurrences of a challenge in literature.

5 Discussion

This section provides a discussion of the results, the threats to validity and future work.

5.1 Discussion of the results

5.1.1 The different dimensions of the NoOps definition and advantages

As presented in the previous section, there are multiple dimensions for the NoOps definition, meaning that no consensus exists in the literature. Although the most popular perspective is that NoOps is a set of practices that is complementary to DevOps, other perspectives show that the key component of the NoOps definition is, clearly, the high level of automation, as well as the abstraction of the IT environment from the underlying infrastructure. When it comes to the necessity of a dedicated IT Operations team, the opinions diverge. While some claim that NoOps could replace IT Operations teams completely, others believe that IT Operations are still needed, although the team size could be reduced. Based on these dimensions, the following definition can be formulated:

NoOps is a set of technical and cultural practices that promotes highly automated IT environments which are abstracted away from the underlying infrastructure, aiming at reducing or even eliminating the necessity of in-house operations teams, but also the involvement of developers in operational activities.

15

Regarding the benefits that the NoOps model brings, these were categorized into four categories: Operational Processes, Social, Technical, and Organizational and Business. Operational Processes benefit the most from NoOps, which seems to minimize the risk of human error, enhance reliability, lower the risk for cyber-attacks and by enabling self-service. In addition, the adoption of NoOps reduces the need for incident responses due to the additional intelligence brought into the operations processes and increases the operational efficiency. From a social perspective, NoOps is expected to improve collaboration between teams, to increase accountability, provide learning opportunities for IT Operations teams and to allow developers to create value by reducing the need to interoperate with IT Operations teams. From a technical perspective, NoOps provides opportunities for integrating quality engineering and testing automation, but it also provides scalability of processes, since many manual activities are replaced by automation. Furthermore, the automation of operational activities requires developers to write high quality code that satisfies the automation requirements. In other words, NoOps practices could increase the code quality. For the IT organization where NoOps would be adopted, the model is expected to reduce operational and infrastructure costs, reducing the need for scaling the operations teams. Moreover, NoOps could help organizations build in governance, observability, quality and compliance.

5.1.2 The characteristics of NoOps

As observed in the literature, the main pillar of NoOps from a technical perspective is automation. The automation aspects that practitioners have associated with NoOps are automated testing, auto-scaling, build automation, automated security scans, infrastructure provisioning, reporting of critical information, deployments, monitoring, as well as management of applications. The most frequent aspect, however, is automated monitoring, mainly due to the fact that monitoring is a tedious activity, especially when it is performed manually. From an architectural perspective, it has been reported that the microservice architecture is best suited for integration with NoOps practices, mainly due to the modularity of the applications, which facilitates the operational processes. Regarding NoOps tooling, practitioners reported the use of Continuous Integration tools, virtual assistants for operational activities, software bots, self-service portals, as well as (container) orchestration mechanisms. Finally, NoOps has often been associated to a number of practices. While some of these practices are general, such as Continuous Delivery, quality engineering and enabling the support for Site Reliability Engineering, others could be considered specific to NoOps. The NoOps-specific practices include the use of Everything-as-Code models (such as Infrastructure-as-Code), self-healing software, self-diagnostics, proactive monitoring, the AI/ML-based automation of operations, chaos engineering, as well as outsourcing operational activities to Serverless/cloud providers.

From a cultural perspective, four main types of characteristics have been identified: collaborative, organizational, mindset and responsibility shifting. The collaborative aspects of NoOps involve the removal of the historical silo between development and ITOps teams, removing the walls between IT and business, as well as the transfer of skills between operations and development teams. It is important to note that DevOps strongly advocates the same collaboration principles, which could potentially mean that the establishment of a DevOps culture could be a prerequisite for NoOps. From an organizational perspective, NoOps is characterized by the involvement and acceptance of management into the adoption process, as well as the continuous training of teams on new automation tools. From a mindset perspective, NoOps requires a proactive mindset, engineers being required to understand the variability of operations and design solutions with this variability in mind. In addition, reproducibility of processes is a key component of the NoOps mindset. Finally, responsibility shifting is one of the main pillars of the NoOps culture. In particular, NoOps involves shifting responsibility for quality assurance and security operations from the production phase to the development phase, as well as shifting the monitoring, deployment and operation responsibilities from IT Operations teams to the software development and architecture teams.

5.1.3 The challenges of NoOps

From a technical perspective, the adoption of NoOps exposes a few main concerns. First of all, due to the outsourcing of operations to third-parties (e.g. Serverless or cloud service providers), practitioners reported problems such as vendor lock-in or technical limitations over which they do not have any control. However, another significant issue is the fact that NoOps might not be ideal for any system. Although NoOps adoption is significantly facilitated for cloud-native systems, the situation is different in the case of legacy systems, where re-engineering the system would often be required.

16

As a cultural challenge, NoOps requires a significant mindset shift for developers, who need to ensure that the developed artifacts have an identical behavior in both the local and production environments. From a social perspective, it has been reported that the adoption of NoOps led to reported discomfort in the initial adoption phases, practitioners facing reported hesitation towards understanding the business requirements.

5.2 Threats to validity

This section will present the threats to validity that have been identified for this study, based on the validity criteria identified by Easterbrook et al. [20].

External validity refers to the extent to which the resulted findings can be generalized. In the context of the presented results, considering that theoretical saturation has not been reached, it can not be guaranteed that these results are completely representative for the concept of NoOps in its entirety. In addition, since the study also involved the analysis of audio and video resources, there is the risk that some information could have been missed during the transcription process, especially due to the chunk transcription method that has been used.

Internal validity, as described by Easterbrook et al. [20] refers to whether the results that emerged from data are valid. One of the main risks is the fact that the analysis process was based on content analysis methods (e.g. open coding and card sorting), which are highly subjective. In other words, the presented results could be just a reflection of the author's beliefs. Since these results have not yet been verified by a domain expert and since no triangulation methods have been applied, then this represents a threat to internal validity. Another threat is the fact that theoretical saturation has not been achieved. Although 3 batches of 10 primary studies have been analyzed, it is likely that more insights could emerge if more primary studies would have been analyzed.

Construct validity refers to whether the adopted research methods and the considered variables have been applied and interpreted correctly. One potential risk to construct validity is represented by the lack of thoroughness of the used search strings. Since not all synonyms for the main keywords of the research questions have been considered in the creation of the search strings, it can not be fully guaranteed that the obtained search results are the only ones that could have been relevant to our research questions. In addition, although some research questions explicitly refer to the perspective of practitioners, for some primary studies we have not been able to find whether the author is a practitioner. Hence, the findings could reflect the perspectives of a non-practitioner, which would contradict some of the research questions.

Finally, reliability is defined as whether the results obtained in this study can be replicated by other researchers. In order to increase the reliability of the study, a replication package has been created, containing all the data and all the procedures that have been used to derive the conclusions [1]. In addition to this, since subjective approaches have also been used in some parts of the study, we have considered the computation of inter-rater reliability measures in order to investigate the degree of subjectivity. Although discussions about inter-rater reliability measures existed, due to lack of time we have not been able to conduct the measurements.

5.3 Future work

As previously mentioned, the current version of the study analyzed only 30 resources out of 98, theoretical saturation not being achieved yet. Therefore, it is of utmost priority to continue with the analysis of additional resources, in order to achieve this theoretical saturation and provide a better overview of NoOps.

Another opportunity for future work is to investigate whether the differences in the presented findings are related to the source of information. Since these results originate from a wide variety of sources, it could be assumed that in the case of companies, for example, the information could be biased more towards the services provided by the company. For instance, it could be assumed that a company that offers cloud consultancy services would be more in favor of Serverless solutions for operations.

As previously discussed, some characteristics of NoOps seem to be strongly related to DevOps. In addition, NoOps has often been positioned as complementary to DevOps. Therefore, another direction for future work

[1] The replication package can be found here: `https://drive.google.com/drive/folders/1vOxpos-1KOMOCksEXmGFv7bZ_LOQr_AY?usp=sharing`

would be to investigate how NoOps is positioned relative to DevOps in terms of characteristics.

An orthogonal direction for future work could be to further investigate the identified challenges, and study what kind of solutions have already been proposed, if any.

6 Conclusions

The current study represents a systematic literature mapping aiming at obtaining a preliminary overview of the main perspectives regarding the NoOps concept, as reported by practitioners. The main difficulty of this study is the use of gray literature, which is motivated by the lack of peer-reviewed literature on the topic. The focus on gray literature proved to be challenging, especially due to the questionable quality of the obtained resources, which could not be assessed in a complete and objective manner.

The study provides an overview of the main dimensions of the NoOps definition, the advantages reported by practitioners, as well as the technical and cultural characteristics that have been commonly associated to the concept. Lastly, a number of challenges related to NoOps have been reported, most of them being of technical nature.

References

[1] "What is the traditional waterfall model and why does it fail?," Nov 2020. https://geekbot.com/blog/what-is-the-traditional-waterfall-model-and-why-does-it-fail.

[2] V. Farcic, "Software development models: Waterfall," Oct 2016. https://technologyconversations.com/2014/01/10/software-development-models-waterfall.

[3] R. Lynn, "History of agile: Planview leankit," Nov 2019. https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/history-of-agile/.

[4] A. Rivera, "Devops didn't exist when i started as a developer: How this one principle changed my career," Sep 2019. https://circleci.com/blog/devops-did-not-exist/.

[5] Atlassian, "What is devops?." https://www.atlassian.com/devops.

[6] E. Freeman, "Devops," 2019. https://aws.amazon.com/devops/what-is-devops/.

[7] M. Gualtieri, S. Carielli, and C. McKinnon, "I don't want devops. i want noops.," Nov 2011. https://go.forrester.com/blogs/11-02-07-i_dont_want_devops_i_want_noops/.

[8] J. Schöpfel, "Towards a Prague definition of grey literature," *The Grey Journal (TGJ): An international journal on grey literature*, vol. 7, January 2011.

[9] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, vol. 106, pp. 101 – 121, 2019.

[10] V. Garousi, M. Felderer, M. V. Mäntylä, and A. Rainer, "Benefitting from the grey literature in software engineering research," in *Contemporary Empirical Methods in Software Engineering* (M. Felderer and G. H. Travassos, eds.), pp. 385–413, Cham: Springer International Publishing, 2020.

[11] A. Yasin, R. Fatima, L. Wen, W. Afzal, M. Azhar, and R. Torkar, "On using grey literature and google scholar in systematic literature reviews in software engineering," *IEEE Access*, vol. 8, pp. 36226–36243, 2020.

[12] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, June 2008.

[13] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," vol. 2, January 2007.

[14] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications, 2004.

[15] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[16] V. Alves, N. Niu, C. Alves, and G. Valença, "Requirements engineering for software product lines: A systematic literature review," *Information and Software Technology*, vol. 52, no. 8, pp. 806 – 820, 2010.

[17] C. Bird, "Interviews," in *Perspectives on Data Science for Software Engineering* (T. Menzies, L. Williams, and T. Zimmermann, eds.), pp. 125 – 131, Boston: Morgan Kaufmann, 2016.

[18] S. Wagner and D. M. Fernández, "Analysing text in software projects," *CoRR*, vol. abs/1612.00164, 2016. http://arxiv.org/abs/1612.00164.

[19] T. Zimmermann, "Card-sorting: From text to themes," in *Perspectives on Data Science for Software Engineering* (T. Menzies, L. Williams, and T. Zimmermann, eds.), pp. 137 – 141, Boston: Morgan Kaufmann, 2016.

[20] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, “Selecting empirical methods for software engineering research,” in *Guide to Advanced Empirical Software Engineering* (F. Shull, J. Singer, and D. I. K. Sjøberg, eds.), pp. 285–311, London: Springer London, 2008.

Appendices

A Search results statistics

Search engine	Search string	# raw results
Bing	No Ops	200
	No Ops culture	20
	No Ops definition	10
	No Ops practices	2
	No Ops challenges	0
	No Ops characteristics	0
	No-Ops	200
	No-Ops culture	20
	No-Ops definition	6
	No-Ops practices	2
	No-Ops challenges	0
	No-Ops characteristics	0
	NoOps	194
	NoOps challenges	9
	NoOps culture	50
	NoOps definition	45
	NoOps practices	30
	NoOps characteristics	0
Google	No Ops	191
	No Ops culture	0
	No Ops definition	0
	No Ops practices	0
	No Ops challenges	0
	No-Ops	202
	No-Ops culture	23
	No-Ops definition	9
	No-Ops practices	7
	No-Ops challenges	0
	No-Ops characteristics	2
	NoOps	200
	NoOps challenges	21
	NoOps characteristics	14
	NoOps culture	55
	NoOps definition	38
	NoOps practices	28
Google Scholar	No Ops	200
	No Ops culture	0
	No Ops definition	0
	No Ops practices	0
	No Ops challenges	0
	No-Ops	200
	No-Ops culture	0
	No-Ops definition	0
	No-Ops practices	0
	No-Ops challenges	0
	No-Ops characteristics	0
	NoOps	200
	NoOps challenges	0

	NoOps characteristics	0
	NoOps culture	2
	NoOps definition	2
	NoOps practices	0
Total Google	790	
Total Bing	788	
Total Google Scholar	604	
Total	2182	

22

B List of resources that are used as primary studies

ID	Title	URL	Author	Date	Resource type	Publisher type
R1	How to get from DevOps to NoOps: 5 steps | The Enterpris	https://enterprise	Anita Engleder	13-Mar-2020	Whitepaper	Community
R2	From NoOps to AllOps - Test Double | Our Blog	https://blog.testd	Jason Grosz	26-Nov-2019	Whitepaper	Company
R3	Moving from DevOps to NoOps with a Microservice Archite	https://www.ibm.	Christopher Ham	29-Jun-2016	Whitepaper	Company
R4	The Evolution of NoOps- Whitepaper on Driving More Valu	https://www.mind	Mindtree	1-Oct-2020	Whitepaper	Company
R5	Quality for DevOps teams	https://books.goo	Rik Marselis	1-Mar-2020	Book	Book
R6	No-Ops: How a DevOps strategy centered on automation c	https://www.capg	Venky Chennapr	14-May-2020	Whitepaper	Company
R7	Autonomous Cloud Enablement aka Scaling NoOps via Se	https://www.dyna	Andreas Grabner	7-Feb-2020	Whitepaper	Company
R8	NoOps, DevSecOps, and Managing Cyber Risk - …	https://deloitte.ws	Ken Corless, Mik	27-Feb-2019	Magazine article	Magazine
R9	Digital transformation: How we took NoOps to the …	https://enterprise	Bernd Greifened	27-Feb-2020	Whitepaper	Community
R10	wary of the economics of" Serverless" Cloud Computing	https://ieeexplore	Adam Eivy	26-Apr-2017	Paper	Scientific journal
R11	Introducing a no-ops culture to remove bottlenecks from cl	https://www.yout	Mike Ensor	14-Jun-2019	Video	Company
R12	Is NoOps the End of DevOps? Think Again | Blog | AppDyr	https://www.appc	Jordan Bach	11-Apr-2017	Blog article	Company
R13	NoOps in a serverless world | Deloitte Insights	https://www2.del	Ken Corless, Mik	16-Jan-2019	Blog article	Company
R14	What is NoOps? Is it Agile Ops? - Botmetric	https://www.botm	Jayashree Hegde	25-May-2017	Whitepaper	Company
R15	DevOps Ventured, NoOps Gained: The Natural Progressio	https://www.fusio	Harbinder Kang	2-May-2019	Whitepaper	Company
R16	NoOps - Is this the end of DevOps as we know it? - - Strat	https://www.strat	Rotem Dafni	31-Oct-2016	Blog article	Company
R17	[PDF] white paper: demystifying noops and serverless com	https://www.cio.g	Chief Information	1-Jan-2020	Whitepaper	Government
R18	NewOps: The Future of Operations in a Hybrid IT World | A	https://www.avne	Jason Bloomberg	26-Mar-2019	Whitepaper	Community
R19	What is NoOps and How to Switch from DevOps to NoOps	https://smartface	Smartface	11-Apr-2018	Whitepaper	Company
R20	How to Create a Successful NoOps Team | xMatters	https://www.xmat	Adam Serediuk	12-Nov-2015	Whitepaper	Company
R21	Beware the siren song of no-ops | InfoWorld	https://www.infow	David Linthicum	19-Feb-2019	Magazine article	Magazine
R22	The Perspective: NoOps Through the Eyes of a DevOps E	https://www.botm	Jayashree Hegde	28-Jan-2017	Blog article	Company
R23	NoOps – The end of DevOps? – KENOPSY	https://kenopsy.c	Kenopsy Service	14-Apr-2020	Whitepaper	Company
R24	Software Factory #2: What makes infrastructure crucial - W	https://witekio.co	Tangui Colin	1-Sep-2017	Whitepaper	Company
R25	framework for managing mission needs, compliance, and t	https://ieeexplore	B.S. Farroha	6-Oct-2014	Paper	Conference
R26	Why culture is more important than tech adoption in a softv	https://enterprise	Bernd Greifened	20-Jun-2019	Blog article	Community
R27	Redefining NoOps to Better Inform IT Decision-Making	https://devops.co	Hussein Badakh	30-Aug-2017	Blog article	Community
R28	Serverless Computing: Moving from DevOps to NoOps	https://devops.co	Laurent Bride	25-Jun-2018	Blog article	Community
R29	Logging and NoOps with Christian Beedgen - Software En	https://softwaree	Christian Beedge	11-Apr-2016	Podcast	Community
R30	Tech Debate: DevOps vs No Ops - Sysco LABS	https://syscolabs	Sysco Labs	20-Mar-2018	Video	Company
R31	Podcast: DevOps to NoOps: State of Play | Oracle ...	https://blogs.orac	Bob Rhubart	19-Sep-2018	Podcast	Company
R32	Cloud Thought Leadership Series – Part 4: Going …	https://www.mind	Mindtree	1-Dec-2020	Video	Company
R33	What is NoOps? The quest for fully automated IT operatior	https://www.cio.c	Mary K. Pratt	11-Jul-2017	Magazine article	Magazine
R34	How NoOps Improves Development Productivity and Oper	https://blog.engin	Engine Yard	1-Sep-2020	Blog article	Company
R35	Is NoOps Achievable? - Lumen	https://blog.lume	Scott Brindamou	15-Jul-2020	Blog article	Company
R36	A NoOps state of mind - SD Times	https://sdtimes.c	Christina Cardoz	13-Mar-2019	Magazine article	Magazine
R37	NoOps: How serverless architecture introduces a third moc	https://www.tech	Keith Townsend	19-Jan-2018	Blog article	Magazine
R38	NoOps: Importance and its Benefits in IT Operations | E-SF	https://www.e-sp	E-Spincorp	28-Mar-2019	Blog article	Company
R39	Look mum, NoOps! How to empower the next evolution of	https://www.itpro	Michael Allen	5-May-2020	Blog article	Community
R40	From DevOps to NoOps – how to automate your chocolate	https://www.yenl	Ruben van der Z	7-May-2017	Blog article	Company
R41	Your Cloud Management Team: Ops, DevOps or NoOps ?	https://iamondem	Ofir Nachmani	26-Dec-2011	Blog article	Magazine
R42	Ops, DevOps and PaaS (NoOps) at Netflix - Adrian Cockc	http://perfcap.blo	Adrian Cockcroft	19-Mar-2012	Blog article	Ad-hoc blog
R43	The Road to NoOps: Serverless Computing is Quickly Gair	https://thenewsta	Mark Boyd	18-May-2016	Magazine article	Magazine
R44	DevOps, DataOps, GitOps, NoOps ... let's clarify! - Mia Pla	https://blog.mia-p	Mia-Platform Tea	12-Feb-2020	Blog article	Company
R45	NoOps, AppOps, DevOps, & More - Removing the OS Bar	https://blog.newr	Adron Hall	8-Feb-2012	Blog article	Company
R46	NoOps – Reality or just another fad | Brillio Technologies	https://www.brillic	Siva Perubotla	12-Mar-2019	Blog article	Company
R47	No-Ops: It's Been 5 Years - DevOps.com	https://devops.co	Don Macvittie	1-Dec-2017	Blog article	Community
R48	Serverless Superheroes: Adam Johnson, Monitoring, and t	https://www.goog	Forrest Brazeal	13-Sep-2017	Blog article	Community
R49	The blueprints of a “no-ops” startup | Hacker Noon	https://hackernoc	Tal Bereznitskey	9-Aug-2017	Blog article	Community
R50	Netflix uses lots of cloud services -- but don't call it 'NoOps	https://www.comp	Ellen Messmer	27-Mar-2012	Magazine article	Magazine
R51	NoOps is a No-No. In my previous post about the evolution	https://medium.c	Colin But	21-May-2018	Blog article	Community
R52	DevOps Is About Collaboration; NoOps Is About …	https://go.forreste	Mike Gualtieri	29-Jun-2011	Blog article	Company
R53	SDLC Series: NoOps Primer - Polyrific	https://www.polyr	Team Polyrific	2-Jun-2018	Blog article	Company
R54	What is BizDevOps? — DevOps Institute	https://devopsins	Eveline Oehrlich	9-Sep-2019	Blog article	Company
R55	Cherre	https://blog.cherr	Stephan Thorpe	12-Oct-2020	Blog article	Company
R56	Reply to http://perfcap.blogspot.com/2012/03/ops …	https://gist.github	John Allspaw	20-Mar-2012	Blog article	Community
R57	The Role of Operations in a No-Ops World	https://devops.co	Don MacVittie	28-Nov-2017	Blog article	Community
R58	Why Going Serverless Doesn't Mean 'No Ops' – The New	https://thenewsta	Kiran Oliver	20-Jan-2017	Podcast	Magazine
R59	There's no ops like NoOps: the next evolution of DevOps |	https://www.zdne	Joe McKendrick	23-Feb-2019	Blog article	Community
R60	Evolution of NoOps from DevOps - CloudHedge	https://cloudhedg	Eshan Sarpotdar	19-Mar-2019	Blog article	Company
R61	I Don't Want DevOps. I Want NoOps. - Forrester	https://go.forreste	Mike Gualtieri	8-Feb-2011	Blog article	Company
R62	What Is AIOps, BizDevOps, CloudOps, DevOps, ITOps, No	https://www.stxne	Adam Stempniak	25-Apr-2019	Blog article	Company

R63	What is NoOps and Why it is the Future of the IT Industry	https://insights.da	Devi Singh	15-Sep-2020	Blog article	Company
R64	DevOps or NoOps: Can You Have Too Much Automation?	https://www.tech	Bob Aiello	19-Nov-2015	Blog article	Community
R65	Are You Ready for Serverless Computing and NoOps? - W	https://worthwhile	Dan Rundle	22-Aug-2019	Blog article	Company
R66	A Guide to ITOps, DevOps, and NoOps Concepts | NCube	https://ncube.con	Alex Melnichuk	25-Sep-2020	Whitepaper	Company
R67	The 2017 Cloud Trends — From DevOps to NoOps | by Vij	https://articles.mi	Vijay Rayapati	8-Feb-2017	Blog article	Community
R68	DevOps to NoOps A Journey worth taking for Indian Banks	http://bwcio.busir	Maha Santaram	13-May-2019	Blog article	Magazine
R69	Forget DevOps — Is the future of cloud NoOps? - FedScoo	https://www.feds	Carten Cordell	29-Jun-2018	Magazine article	Magazine
R70	Why 2013 is the year of 'NoOps' for programmers [Infograp	https://gigaom.cc	Derrick Harris	31-Jan-2012	Blog article	Company
R71	No-Ops? New-Ops? Cloud-Ops! - Evolven	https://www.evol	Martin Perlin	3-Apr-2012	Blog article	Company
R72	The road to serverless maturity: Running away from "NoOp	https://jaxenter.c	Gabriela Motroc	12-Feb-2018	Magazine article	Magazine
R73	NoOps: Its Meaning and the Debate around It - InfoQ	https://www.infoc	Abel Avram	16-Mar-2012	Blog article	Community
R74	Dynatrace Gets Hands-On With Hands-Off 'NoOps' …	https://www.forbe	Adrian Bridgwate	11-Dec-2019	Magazine article	Magazine
R75	NoOps a no-brainer? Completely autonomous …	https://www.zdne	Joe McKendrick	27-Feb-2019	Blog article	Community
R76	DevOps Trends | Towards Data Science	https://towardsda	Daniele Fontani	20-May-2020	Blog article	Community
R77	[Webinar Recap] NoOps? Or Yes, Ops! The Future …	https://blog.opsra	Aaditya Aravamu	21-Nov-2018	Blog article	Company
R78	What Is NoOps?	https://www.linke	Agustin Romano	19-Jul-2019	Blog article	Community
R79	Serverless Architectures - Martin Fowler	https://www.mart	Mike Roberts	22-May-2018	Blog article	Ad-hoc blog
R80	Keeping NoOps from going rogue	https://devops.co	Don Dingee	29-Apr-2019	Blog article	Community
R81	NoOps' Debate Grows Heated	https://www.cio.c	Nancy Gohring	20-Mar-2012	Magazine article	Magazine
R82	What is NoOps?	https://searchitop	Margaret Rouse	1-Jul-2015	Blog article	Community
R83	NoOps & The Future Of QA Automation (with Lewis Presc	https://theqalead	Lewis Prescott	18-Nov-2020	Podcast	Community
R84	DrupalCon Nashville 2018: Welcome to NoOps, the new D	https://www.gooc	Kieron Sambrook	12-Apr-2018	Video	Conference
R85	DevOps vs NoOps - Criticalcase	https://www.critic	Irene Maida	18-Jan-2018	Blog article	Company
R86	When DevOps isn't enough, try NoOps - Computer Weekly	https://www.com	Clive Longbotton	27-Jul-2015	Blog article	Community
R87	Shifting It's Focus from Operations to Outcomes With NoO	https://enterprise	Meeta Ramnani	17-Jul-2019	Blog article	Community
R88	[PDF] Is NoOps the end of the road for operations professi	https://www.brigh	Brightred	5-Apr-2019	Whitepaper	Company
R89	NoOps - A Big Lie or a Political Shift? - SiliconANGLE	https://siliconang	Alex Williams	1-Feb-2012	Blog article	Magazine
R90	Oops, Netflix: NoOps is a No Go (NASDAQ:NFLX, NASDA	https://www.chea	Dlallah Haidar	28 March 2012	Magazine article	Magazine
R91	The Emotional Pull of NoOps | Zenoss	https://www.zenc	Robyn Weisman	28-Feb-2018	Blog article	Company
R92	2019 Deloitte tech trends predictions: AI-fueled firms, NoO	https://hub.packt	Natasha Mathur	29-Jan-2019	Blog article	Community
R93	Is NoOps Killing DevOps- Experts Disagree - Codelattice E	https://blog.codel	Vijith Sivadasan	5-Dec-2018	Blog article	Company
R94	Say 'yes' to NoOps with the right technology and …	https://searchitop	Will Kelly	26-Nov-2019	Blog article	Community
R95	Cloud computing ushers in a new era for DevOps: …	https://searchsof	Crystal Bedell	11-Jul-2011	Blog article	Community
R96	Should NoOps be your new operations strategy? - …	https://medium.c	Vishrutha Amuda	24-Jun-2019	Blog article	Community
R97	Blog | NoOps! Is That Even A Term? | GS Lab	https://www.gslal	Ravindra Yadav	24-Apr-2018	Blog article	Company
R98	No-Ops – Next Evolution or Fuss? - Words Geek	https://wordsgeel	Ashit Mithal	20-Sep-2020	Blog article	Ad-hoc blog

24

C List of codes for each research question

RQ1. NoOps definition	NoOps implies that an IT environment is highly automated, such that there is no necessity for a dedicated team to manage operations in-house NoOps implies total exclusion of IT Ops NoOps implies that an IT environment is highly automated and abstracted from the underlying infrastructure, such that there is a reduced necessity for a dedicated team to manage operations in-house NoOps implies that an IT environment is highly automated and abstracted from the underlying infrastructure, such that there no necessity for a dedicated team to manage operations in-house NoOps is the mindset that no traditional operational tasks are necessary to deploy and operate software NoOps implies that an IT environment is highly automated, such that there is a reduced necessity for a dedicated team to manage operations in-house NoOps implies that developers do not have to interact with the operations team NoOps is complementary to DevOps NoOps is the goal of completely automating the deployment, monitoring, management of applications and infrastructure
RQ1. NoOps advantages	Provides scalability Facilitates testing automation Lowers the risk for cyberattacks Reduces necessity for scaling ITOps team size Improves collaboration Enables self-service Increases accountability Build in Governance, Observability, Quality and Compliance Enhances operations process with further intelligence, reducing the need for incident responses Facilitates quality engineering Enhances reliability More time and opportunities for learning and growth for ITOps team Minimizes the risk of human error Cost savings for infrastructure and application operations Eliminates friction with the infrastructure operations (especially in the release phase), such that developers can create value by not having to interoperate with operations teams Additional operational efficiency for infrastructure and application operations Incentivizes developers to write better quality code

Table 13: Coding results for research question 1

RQ2. Technical characteristics	Cost-benefit analysis for deciding which aspects to automate
	Quality engineering
	Virtual assistants
	Support for Site Reliability Engineering
	Orchestration mechanisms
	Continuous Integration tooling
	Automated security scans
	Microservice architecture suitability
	Automation of test-release-operate phases
	Build automation
	Automated testing
	Automated reporting of critical information
	Software bots
	Continuous Delivery
	Ensuring service observability
	Automated infrastructure provisioning
	Self-diagnostics
	Implementing feedback cycles
	Auto-scaling
	Automated management of applications
	Self-service portals for operations
	Everything-as-Code
	AI/ML-based automation of operations
	Proactive monitoring
	Automated deployment
	Self-healing
	Automated monitoring
	Outsourcing operational activities to serverless/cloud providers
	Chaos engineering
RQ2. Cultural characteristics	Shifting quality assurance from production stage to development stage
	Shifting monitoring responsibility from the ITOps teams to the development teams
	Shifting security operations from production stage to development stage
	Management involvement and acceptance
	Aiming for process reproducibility
	Responsibility for deployment and operation is shifted to software engineering teams
	Removing the silo between Dev and ITOps teams
	Training teams on new automation tools
	Proactivity in engineering the variability of operations
	Removing the walls between IT and business
	Transfer of skills
	Shifting developers' thoughts towards taking ownership of the platforms
	Shifting responsibility for operations from ITOps teams to Software Architects

Table 14: Coding results for research question 2

RQ3. Adoption challenges - social	Reported discomfort in the initial phases of adoption
	Reported hesitation to understanding the business in the initial phases of adoption
	NoOps can not be based on traditional ways of working together
RQ3. Adoption challenges - technical	Outsourcing activities to third-parties could lead to technical issues and limitations
	Poor NoOps implementation quality could lead to cyber risk at scale
	Adoption may not be suitable or requires re-engineering of legacy systems
	Vendor lock-in
	Management of functions when using FaaS solutions
RQ3. Adoption challenges - cultural	NoOps requires a mindset shift for developers in order to ensure that the developed artifact behaves the same in both local and production environments
RQ3. Adoption challenges - environmental	Regulatory compliance, security and other cross-cutting concerns need to be addressed company-wide
RQ3. General challenges	Adopting NoOps could lead to less involvement from development in operations, which completely contradicts DevOps practices

Table 15: Coding results for research question 3

D Mapping of findings to resources

Primary code	Sub-code	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	Total
RQ1. NoOps definition	NoOps implies that an IT environment is very automated, such that there is no necessity for a dedicated team to manage operations in-house	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	NoOps is the goal of completely automating the deployment, monitoring, management of applications and infrastructure	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	1
	NoOps implies that an IT environment is very automated and abstracted from the underlying infrastructure, such that there is a reduced necessity for a dedicated team to manage operations in-house	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	NoOps implies that an IT environment is very automated, such that there is a reduced necessity for a dedicated team to manage operations in-house	N	N	N	N	Y	N	N	Y	Y	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	4
	NoOps implies that developers do not have to interact with the operations team	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	3
	NoOps implies that an IT environment is very automated and abstracted from the underlying infrastructure, such that there no necessity for a dedicated team to manage operations in-house	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y	N	N	Y	Y	N	Y	N	N	N	N	N	N	5
	NoOps is the mindset that no traditional operational tasks are necessary to deploy and operate software	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	Y	Y	N	N	N	4
	NoOps implies total exclusion of IT Ops	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y	Y	Y	Y	6
	NoOps is complementary to DevOps	N	N	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y	N	N	Y	N	Y	N	N	Y	Y	N	N	N	Y	N	N	N	11
	Provides scalability	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Facilitates testing automation	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Lowers the risk for cyberattacks	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Reduces necessity for scaling ITOps team size	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Improves collaboration	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Enables self-service	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Increases accountability	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Build in Governance, Observability, Quality and Compliance	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1

Primary code	Sub-code	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	Total
RQ1. NoOps advantages	Enhances operations process with further intelligence, reducing the need for incident responses	N	N	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Facilitates quality engineering	N	N	N	Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Enhances reliability	N	N	N	N	Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Incentivizes developers to write better quality code	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	1
	Minimizes the risk of human error	N	N	N	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	More time and opportunities for learning and growth for ITOps team	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	Y	N	4
	Cost savings for infrastructure and application operations	N	N	N	Y	N	Y	N	N	N	N	N	N	Y	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	Y	5
	Additional operational efficiency for infrastructure and application operations	N	N	N	N	N	N	N	N	Y	N	N	N	N	Y	Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	4
	Eliminates friction with the infrastructure operations (especially in the release phase), such that developers can create value by not having to interoperate with operations teams	N	N	N	Y	N	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y	6
RQ2. Technical characteristics	Cost-benefit analysis for deciding which aspects to automate	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Quality engineering	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Virtual assistants	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Support for Site Reliability Engineering	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Orchestration mechanisms	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Microservice architecture suitability	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Chaos engineering	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	1
	Automated security scans	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	2
	Automation of test-release-operate phases	N	N	N	Y	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Automated testing	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	2
	Automated reporting of critical information	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	2
	Software bots	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Continuous Delivery	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	2
	Ensuring service observability	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Continuous Integration tooling	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Build automation	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	Y	N	N	N	N	N	N	N	3
	Self-diagnostics	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1

29

Primary code	Sub-code	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	Total
	Auto-scaling	N	N	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Automated management of applications	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Automated infrastructure provisioning	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Implementing feedback cycles	N	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	3
	Self-service portals for operations	N	N	N	N	N	Y	Y	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	3
	AI/ML-based automation of operations	Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	3
	Proactive monitoring	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	3
	Self-healing	N	N	Y	N	N	N	Y	N	Y	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	4
	Everything-as-Code	N	N	N	N	N	N	N	N	N	N	Y	N	Y	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	4
	Automated deployment	Y	N	N	N	N	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	Y	5
	Automated monitoring	N	N	Y	N	N	N	N	N	N	N	Y	Y	N	Y	Y	N	N	N	N	Y	N	N	Y	N	N	N	N	N	N	N	7
	Outsourcing operational activities to serverless/cloud providers	N	N	N	N	N	N	N	Y	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N	Y	17
RQ2. Cultural characteristics	Aiming for process reproducibility	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Proactivity in engineering the variability of operations	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Removing the walls between IT and business	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Shifting developers' thoughts towards taking ownership of the platforms	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	1
	Shifting responsibility for operations from ITOps teams to Software Architects	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	1
	Shifting monitoring responsibility from the ITOps teams to the development teams	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Shifting security operations from production stage to development stage	N	N	N	N	N	N	N	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Training teams on new automation tools	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Transfer of skills	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y	N	N	N	N	N	N	N	N	Y	N	N	N	N	3
	Shifting quality assurance from production stage to development stage	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Management involvement and acceptance	Y	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Removing the silo between Dev and ITOps teams	N	N	Y	N	N	N	N	N	Y	N	Y	N	Y	N	N	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	6
	Responsibility for deployment and operation is shifted to software engineering teams	N	N	N	N	N	N	N	N	Y	N	Y	N	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	Y	5

Primary code	Sub-code	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	Total
RQ3. Adoption challenges - social	NoOps can not be based on traditional ways of working together	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Reported hesitation to understanding the business in the initial phases of adoption	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	1
	Reported discomfort in the initial phases of adoption	Y	N	N	N	N	N	N	N	Y	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	3
RQ3. Adoption challenges - technical	Management of functions when using FaaS solutions	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	1
	Poor NoOps implementation quality could lead to cyber risk at scale	N	N	N	N	N	N	N	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Vendor lock-in	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Outsourcing activities to third-parties could lead to technical issues and limitations	N	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	2
	Adoption may not be suitable or requires re-engineering of legacy systems	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	N	Y	N	Y	Y	Y	N	N	N	Y	N	N	N	N	N	N	Y	9
RQ3. Adoption challenges - cultural	NoOps requires a mindset shift for developers in order to ensure that the developed artifact behaves the same in both local and production environments	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	2
RQ3. Adoption challenges - environmental	Regulatory compliance, security and other cross-cutting concerns need to be addressed company-wide	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	3
RQ3. General challenges	Adopting NoOps could lead to less involvement from development in operations, which completely contradicts DevOps practices	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	2