

# A Mixed Method Study of DevOps Challenges

Minaoar Hossain Tanzil<sup>1,\*</sup>, Masud Sarker<sup>2</sup>, Gias Uddin<sup>1</sup>, Anindya Iqbal<sup>2</sup>

---

## Abstract

**Context:** DevOps practices combine software development and IT (Information Technology) operations. The continuous needs for rapid but quality software development requires the adoption of high-quality DevOps tools. There is a growing number of DevOps related posts in popular online developer forum Stack Overflow (SO). While previous research analyzed SO posts related to build/release engineering, we are aware of no research that specifically focused on DevOps related discussions. **Objective:** This paper aims to learn the challenges developers face while using the currently available DevOps tools and techniques along with the organizational challenges in DevOps practices. **Method:** We conduct an empirical study by applying topic modeling on 174K SO posts that contain DevOps discussions. We then validate and extend the empirical study findings with a survey of 21 professional DevOps practitioners. **Results:** We find that: (1) There are 23 DevOps topics grouped into four categories: Cloud & CI/CD Tools, Infrastructure as Code, Container & Orchestration, and Quality Assurance. (2) The topic category ‘Cloud & CI/CD Tools’ contains the highest number of topics (10) which cover 48.6% of all questions in our dataset, followed by the category Infrastructure as Code (28.9%). (3) The file management is the most popular topic followed by Jenkins Pipeline, while infrastructural Exception Handling and Jenkins Distributed Architecture are the most difficult topics (with least accepted answers). (4) In the survey, devel-

---

\*Corresponding author

Email address: minaoar@gmail.com (Minaoar Hossain Tanzil)

<sup>1</sup>University of Calgary

<sup>2</sup>Bangladesh University of Engineering and Technology

opers mention that it requires hands-on experience before current DevOps tools can be considered easy. They raised the needs for better documentation and learning resources to learn the rapidly changing DevOps tools and techniques. Practitioners also emphasized on the formal training approach by the organizations for DevOps skill development. **Conclusion:** Architects and managers can use the findings of this research to adopt appropriate DevOps technologies, and organizations can design tool or process specific DevOps training programs.

*Keywords:* DevOps, CI/CD, Jenkins, Infrastructure as Code

---

## 1. Introduction

The term ‘DevOps’ was coined in early 2000 to combine software development and IT (Information Technology) practices with the aim to improve software product development and delivery [1, 2]. According to UpGuard [3], 63% organizations in 2021 experienced improvement in the quality of their software deployment and frequency of new software releases and 55% of the organizations noticed improved collaboration among teams by adopting DevOps tools. Several research studies [4, 5] show that DevOps adoption can improve software quality assurance and help software teams achieve high degree ownership, but such adoption requires steep learning curve for software developers, operation engineers, and the organization as a whole. As found in the 2015 State of the DevOps report [6], DevOps initiatives launched solely by C-level executives or from the grassroots are less likely to succeed, it is important to understand the challenges engineers face while using DevOps tools and practices for a successful DevOps adaptation.

We observe that discussions about various DevOps problems are prevalent in the popular online developer forum Stack Overflow (SO) [7]. Several research has been conducted to analyze SO posts (e.g., big data [8], concurrency [9], blockchain [10], microservices [11]). Recently, SO discussions on modern release engineering are analyzed using topic modeling [12]. While release engineering practices can belong to DevOps practices, the concept of DevOps is broad as

it contains tools and practices around several areas like continuous development, deployment, monitoring, maintenance, etc. As such, we are aware of no research that analyzed DevOps discussions on SO, although such insight can complement existing DevOps literature which so far has mainly used controlled case studies/surveys in the industry [4, 5, 13].

In this paper, we report a mixed method study to understand the challenges DevOps practitioners face. **First**, we conduct an empirical study by applying topic modeling on 174K SO posts related to DevOps discussion (Section 3). We find 23 topics grouped into four categories: Cloud & CI/CD Tools, Infrastructure as Code, Container & Orchestration, and Quality Assurance. The category ‘Cloud & CI/CD Tools’ contains the highest number of topics (10) and the most number of questions (>48%), followed by the category ‘Infrastructure as Code’ with 7 topics and 28.9% of all questions. File management and Jenkins pipeline are the most popular (with most views) topics, while Test Automation and Git CI/CD support are the most difficult topics with highest percentage of questions without an accepted answer. **Second**, we validate and extend the findings of the empirical study using a survey of 21 professional DevOps practitioners (Section 4). We find that the survey participants agreed with our observed topics of DevOps challenges in SO. We also find that organizational training is insufficient to improve DevOps skills. Moreover, experts shared difficulties in version compatibility, and strongly predicted that in future, cloud infrastructure automation will be future trend in DevOps. Our research contribution is summarized in Table 1.

Our study findings can be useful to several stakeholders in Software Engineering (SE). The DevOps developers and practitioners can use the observed topics to prioritize/improve their learning of DevOps tools. DevOps vendors can guide their tool development and SE researchers can develop innovative techniques and documentation to address the challenges.

The paper is outlined as follows: we explore the DevOps concepts and related subdomains in section 2, we present the methodology of empirical study performed on Stack Overflow and also the topics and difficulties found in sec-

Table 1: Summary of research contributions and advancements made by our study

Method	Research Contribution	Research Advancement
Empirical Study	Out of 174K SO posts, we observed 23 DevOps topics under 4 major categories of Cloud & CI/CD Tools, Infrastructure as Code, Container & Orchestration, and Quality Assurance. We also analyzed most viewed (File management and Jenkins pipeline) and most unresolved challenging topics (Test Automation and Git CI/CD support) in SO.	As per our knowledge, no other study analyzed DevOps topics in SO. Previous studies in closely related domain of continuous software engineering and release engineering did not find major DevOps topics that we found, e.g., Kubernetes, Cloud-Infra Automation, Non-functional Test Automation etc. Our study found that Kubernetes, which is a very important part of modern infrastructure orchestration, comprises of 13.4% of all DevOps posts; Cloud-Infra Automation has 6.2% posts; and Non-function Test Automation related posts take the highest amount of time (251 hours) to get a response among all topics. Moreover, no other previous researches found staggering 21.4% of DevOps posts related with Jenkins only. These unique findings provide significant insights to SE researchers and infrastructure and test automation vendors.
Survey	We conducted a practitioners survey to evaluate our empirical findings and to collect people and process related DevOps challenges. 21 DevOps practitioners responded to quantitative and open-ended questions on DevOps tools, practices, challenges, and trends.	We reported that DevOps related skill shortage is major bottleneck faced by practitioners in the industry. Still only 5% of organizations provide formal training for DevOps. As we found prevalence of Cloud-Infra Automation topic in our empirical study, practitioners also re-enforced strong prediction in this future technology trend. These kind of qualitative insights related with people, organization, and technology trend are critical for management and improvement of DevOps practices which would go unnoticed otherwise.
Tag set analysis	At the initial stage of our empirical study, we had to select a final tag-set from 30 different tag-sets. We provided detailed replicable steps how such topic related final tag-set can be generated through manual analysis.	Though it is common in empirical studies to generate a list of relevant tag-sets using some significance and relevance values, we could not find any other study to explain how they choose the final tag-set among dozens of such tag-sets. We have provided step-by-step detail analysis process with examples on how we discarded the non-relevant tags and finally selected the relevant tag-set. Future empirical studies can be significantly benefited to replicate this method.

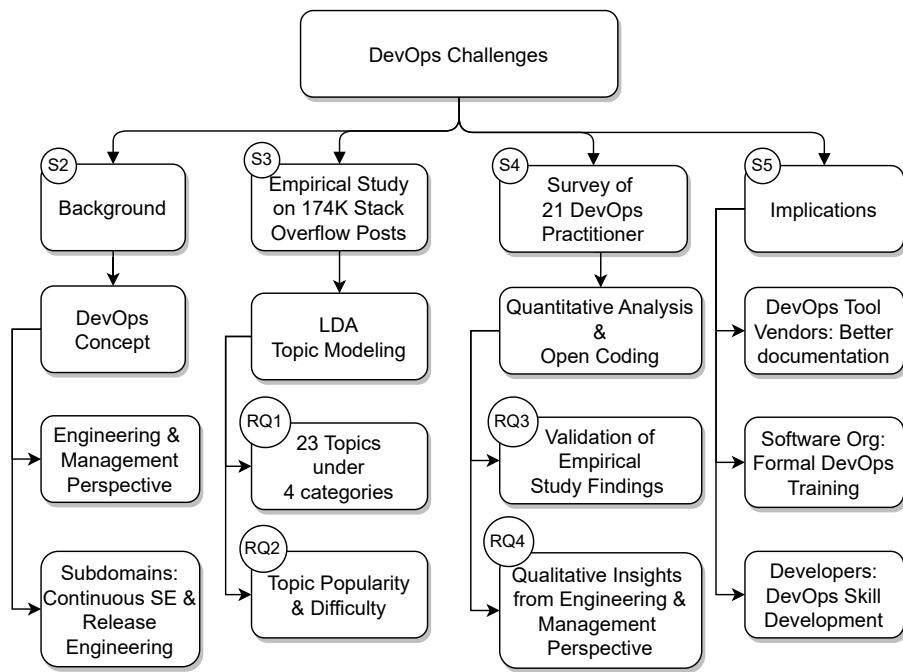


Figure 1: Outline of the paper, describing four major sections and four research questions.

S-n refers to section n, and RQ-n refers to search question n.

tion 3 which is followed by a practitioners' survey design and findings that can validate and extend the empirical results. Finally, implications for the industry and academia in section 5 is followed by threats to validity and related works in section 6 and 7 respectively. The total flow of the paper is outlined in Figure 1.

**Replication Package:** <https://github.com/DevOpsTopic/Challenges> [14].

## 2. Background

**DevOps Concepts.** Aiello et al. defined DevOps in [15] as *DevOps is a set of principles and practices intended to help development and operations collaborate and communicate more effectively*. A slightly extended definition is provided by Leite et al. [13], DevOps “*is a collaborative and multidisciplinary organizational effort to automate continuous delivery of new software updates while guaranteeing their correctness and reliability*”. In this literature survey, Leite et al. [13] identified four major categories of DevOps concepts, namely, Process, People, Delivery, and Runtime using a systematic analysis on 50 ‘core’ DevOps publications till 2019. While process and people are relevant from the management perspective, delivery (development related) and runtime (operation related) are associated from the engineering perspective. The categories are depicted in Figure 2.

The **process** category covers business-related planning and strategies. **People** encompasses concepts regarding skills, incentives and the culture of collaboration. **Delivery** consists of the concepts necessary for continuous build, delivery and releases, and, finally, **Runtime** elaborates concepts related to the guarantee of operational stability and reliability of services in a continuous software engineering environment.

### Continuous Software Engineering Processes in DevOps.

Given the broad conceptual coverage of DevOps, it is closely related with the processes of Continuous Software Engineering (CSE). Fitzgerald et al. extended the concept of CSE and proposed a holistic approach of ‘Continuous \*’,

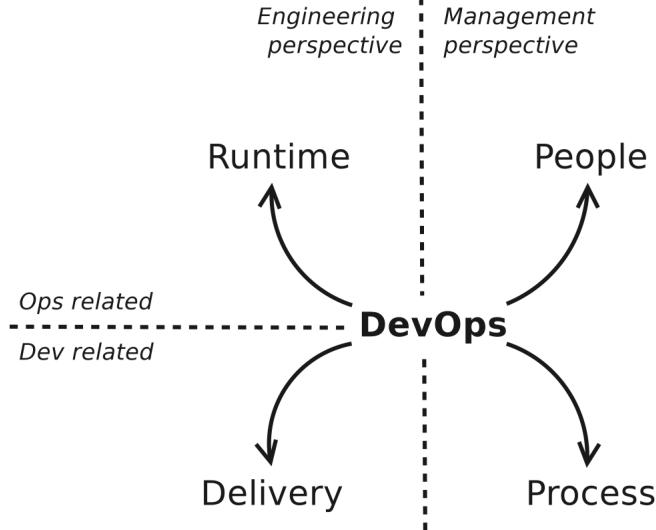


Figure 2: DevOps Conceptual Map provided by Leite et al. [13]

in [16]. They divided the entire software life-cycle into three main sub-phases: Business Strategy and Planning, Development, and Operations. Within these sub-phases, they positioned the various categories of continuous software engineering activities (all-continuous planning, integration, deployment, delivery, verification, testing, compliance, monitoring etc.). Finally, they positioned continuous improvement and innovation as the foundation upon which other 'Continuous \*' activities could be grounded. Comparing this proposition of CSE by Fitzgerald et al. [16] and DevOps conceptual framework by Leite et al. [13], CSE covers three concepts of DevOps, namely Process, Delivery, and Runtime and does not consider the People category of DevOps.

**Release Engineering in DevOps.** As noted by Dyck et al. [17], *release engineering and DevOps terms are often confused, misinterpreted, or used as synonyms*. In their paper, they proposed definitions of Release Engineering (RE) and DevOps to keep them clearly distinct. Adams and McIntosh [18] defined RE as *The release engineering process is the process that brings high quality code changes from a developer's workspace to the end user, encompassing code change*

Table 2: Relation and distinction among DevOps, Continuous Software Engineering (CSE), and Release Engineering (RE).

<b>DevOps</b> (Leite et al. [13])	<b>CSE</b> (Fitzgerald et al. [16])	<b>Release Engineering</b> (Dyck et al. [17])
Process	Continuous Planning	
	Continuous Improvement	
	Continuous Innovation	
Delivery	Continuous Integration	Continuous integration: Building & Testing
	Continuous Testing	Integration: Branching and Merging
	Continuous Verification	Build System
	Continuous Deployment	Deployment
Runtime	Continuous Delivery	Infrastructure-as-Code Release
	Continuous Compliance	
	Continuous Security	
	Continuous Use	
People	Continuous Trust	
	Continuous Run-time	
People	Continuous Monitoring	

*integration, continuous integration, build system specifications, infrastructure-as-code, deployment and release.* With this definition, RE is equivalent to the Delivery category of DevOps as explained by Leite et al. [13].

**Relation among DevOps, Continuous Software Engineering and Release Engineering.** To summarize these closely related but often mixed up concepts, we refer to the Table 2 with the definition and activities of each concept outlined by the respective researches mentioned above.

From the detailed activities, definition, and framework of DevOps, CSE, and RE, it is evident that DevOps is a superset of CSE which is a superset of RE. This final relation is depicted in the Figure 3.

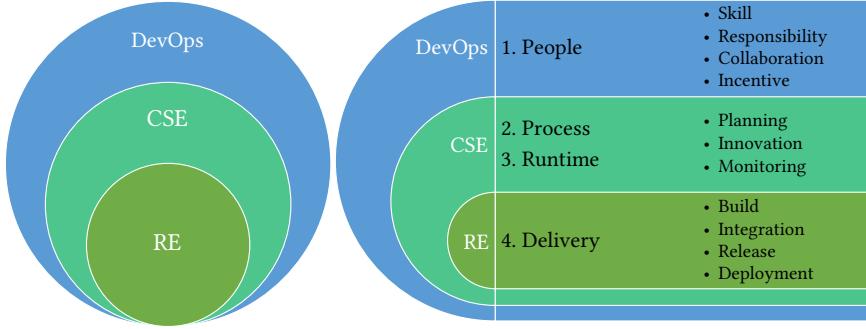


Figure 3: Superset relation of DevOps, CSE, and RE

### 3. Empirical Study of DevOps Discussion

In this section, we apply topic modeling on 174K SO posts related to DevOps discussion and answer two research questions:

**RQ1.** What types of topics are discussed about DevOps in SO?

**RQ2.** How do the popularity and difficulty of DevOps topics vary?

As DevOps is a fast evolving complex process of tools and culture, it is important to recognize the vocabulary and topics in DevOps. In RQ1, we analyze DevOps topics in SO discussions to inform of the challenges around DevOps tools and techniques. All the DevOps topics we observed in RQ1 may not be equally popular or equally difficult. In RQ2, we study of popularity and difficulty of the topics to offer insight into prioritizing research and organizational effort. For example, system architects may decide to choose matured tools with less difficulty or cutting edge tools with active community support [19], organizations may decide which topics need extra investment in terms of employee training. Software engineering researchers and DevOps vendors could also devise ways to make improved tools and techniques using the knowledge.

#### 3.1. Study Data

We collect SO posts in three steps: (1) Download SO dataset, (2) Develop DevOps tag set, and (3) Extract posts using tag set.

### 3.1.1. Download SO Dataset

We downloaded the SO dataset which is publicly available in SOTorrent [20]. The dataset includes SO question and answer posts and their metadata. The metadata includes post identifier, its type (question or answer), title, body, tags, creation date, view count, score, favorite count, and the identifier of the accepted answer (if the post is a question). The collected dataset includes 49,139,907 questions and answers posted over the last 13 years from August 2008 to June 2020.

### 3.1.2. Develop DevOps Tag Set

In this step we develop a set of DevOps tags from a initial tag  $T_{init} = \text{DevOps}$ . Then from our dataset ( $Q$ ) we extract the question posts ( $P$ ) whose tags match a tag in  $T_{init}$ . From the initial posts ( $P$ ), we extract the candidate tag set ( $t$ ). The candidate tags are evaluated by three criteria A, B, and C. In particular, 'A' denotes the number of posts from  $P$  for each of the candidate tags  $t$  (number of posts of questions whose tag contains  $t$  and "DevOps"). 'B' denotes the number of all posts in  $Q$  for each of the candidate tags  $t$  (number of posts in the whole dataset for each of the candidate tags  $t$ ). Additionally, 'C' denotes the number of posts in  $P$ . Tags are then refined further by their relevance. Two heuristics ( $\mu$ ) and ( $v$ ) are used to measure from previous work [21, 22, 23] and relevance of a candidate tag  $t$  in the DevOps tag set ( $t$ ).

$$\text{Significance}(\mu) = \frac{(A) \text{ Number of posts with tag } t \text{ in } P}{(B) \text{ Number of posts with tag } t \text{ in } Q}$$

$$\text{Relevance}(v) = \frac{(A) \text{ Number of posts with tag } t \text{ in } P}{(C) \text{ Number of question in } P}$$

A candidate tag  $t$  is significantly relevant to 'DevOps' if its  $\mu$  and  $v$  are higher or equal to certain thresholds. In our study we evaluate different range of values for significance and relevance where  $\mu = \{0.005, 0.008, 0.010, 0.020, 0.030, 0.050\}$  and  $v = \{0.005, 0.010, 0.015, 0.020, 0.025, 0.030\}$ . We collected tag sets for 30 combinations of relevance and significance thresholds so that every tag set contains at least two tags (for significance 0.02, 0.03, 0.05, there were less

than two tags with some relevance thresholds). Out of these 30 candidate tag sets, maximum 42 tags produced maximum 381,675 posts with the minimum threshold of significance (0.005) and relevance (0.005). The authors inspected each tag set manually to identify the tags which are most relevant with DevOps and discarded the tag sets which included tags not specific to DevOps. For example, the largest tag set of 42 tags contained tags bitbucket, sonarqube, digital-ocean, artifactory etc. which can be generic tools or applications not specifically related with DevOps.

The target of manual analysis was to select a tag-set which has the highest number of DevOps related tags. With this aim, first two authors started reviewing all the tag-sets ( $TS_1, TS_2, \dots, TS_{30}$ ) together, starting from the tag-set with highest number of tags. Among the thirty selected tag-sets, the top 5 sets  $TS_1, TS_2, TS_3, TS_4, TS_5$ ,  $TS_6$  had 42, 30, 28, 24, and 19 tags respectively. First two authors jointly reviewed each tag-set to identify any non-DevOps related tag and recorded such tags against each set. For example, if we denote non DevOps related tags of tag-set-N as  $XTS_N$ , then first two authors identified following non-DevOps related tags in each tag-set  $XTS_1 = \text{bitbucket, sonarqube, digital-ocean, amazon-ecs}$ ,  $XTS_2 = \text{amazon-ecs}$ ,  $XTS_3 = \text{sonarqube, cloud}$ ,  $XTS_4 = \text{amazon-ecs}$ . For the tag-set  $TS_5$ , all tags were found to be DevOps related and  $XTS_5 = .$ . Then all four authors reviewed the excluded tags from the top four tag-sets  $XTS_1, XTS_2, XTS_3, XTS_4$  and after a deliberation session together agreed that those tag-sets can be discarded and  $TS_5$  can be the final tag-set since it had the highest number of tags without any non-relevant tag. This joint review in two steps were made possible because the authors started reviewing tag-sets from the largest set in descending order and hence, after reviewing five sets, a final tag-set was decided.

The final tag set  $\delta_{devops}$  with significance,  $\mu=0.008$  and relevance,  $v=0.010$ , contained following 19 tags related with DevOps:

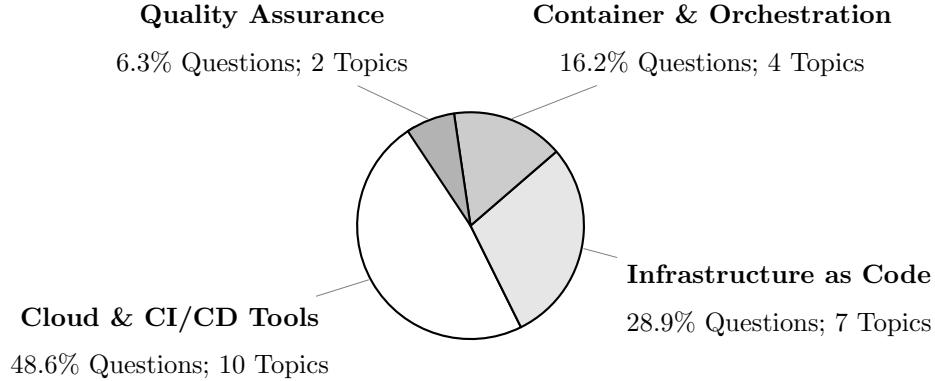


Figure 4: % of questions and # of topics per topic category

**Tag set**  $\delta_{devops} = \{\text{ansible, azure-devops, devops, Jenkins, kubernetes, terraform, chef, continuous-integration, ibm-cloud, gitlab, jenkins-pipeline, gitlab-ci, puppet, amazon-cloudformation, pipeline, azure-pipelines, jenkins-plugins, continuous-deployment, devops-services}\}$

### 3.1.3. Extract DevOps Posts

After generating the DevOps tag set  $\delta_{devops}$ , we extract SO posts whose tag set contain a tag in  $\delta_{devops}$ . Our built dataset from SO includes 174,693 DevOps related question posts and their metadata.

## 3.2. RQ<sub>1</sub> DevOps Topics in Developer Discussion

### 3.2.1. Approach:

Topic modeling has been increasingly used in empirical studies to mine unstructured data [24]; previous works show that topic modeling can generate meaningful topics in software engineering research by mining textual documents [25] [26]. Now we elaborate the process of topic modeling.

Preprocess DevOps Post. We remove irrelevant information and noise from our dataset before applying topics modeling by following noise reduction steps adopted in previous works [8, 27, 28]. We focus only on the post title for removing the noise from our dataset and the preprocessing steps are as follows.

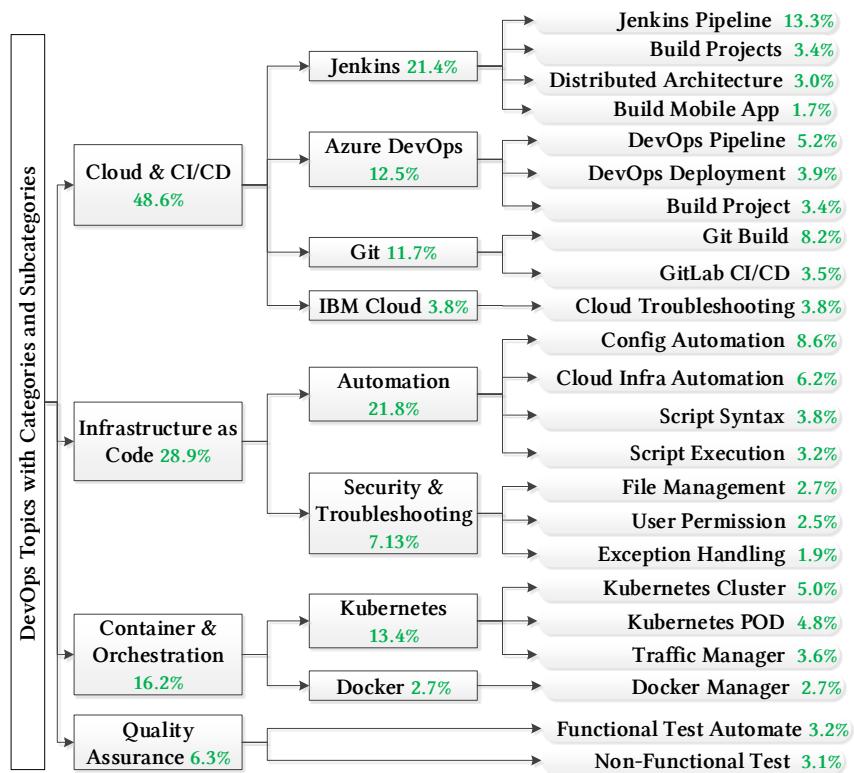


Figure 5: DevOps topics categories and sub-categories

First, we remove the code snippets, HTML tags such as `|p;|/p;, |html;|/html;`, url tags, punctuation, and then stop words (a, an, is, the), numeric and non-alphabetic character by using NLTK stopwords corpus [29]. Besides, bigram models were developed by using Gensim [30] since bigram models enhance the quality of text processing as reported by Tan et al [31]. Moreover, we use Porter stemming to reduce words to their stemmed representation. For example, “configuration”, “configure”, and “configured” all reduce to “config” [32].

Model and Label DevOps Topic. We formulate the topics using Latent Dirichlet Allocation (LDA) [33]. We used MALLET [34] with Latent Dirichlet Assignment (LDA) Topic Modeling to group our DevOps posts into topics. LDA is intended to determine the optimal K-Topic number, whereby LDA generates detailed topics that are difficult to analyze if the topics are too large, but generates overlapping topics if the topics are too small. We ran experiments with a broad range of values  $K = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ , iteration ( $I$ ) =  $\{100, 500, 1000, 1500\}$  and we also set the range of the hyper-parameter values  $\text{alpha} = \{0.05, 0.1, 0.5, 1, 5, 10, 50\}/K$ ,  $\text{beta} = \{0.01, 0.05\}$  and observed the coherence value for each run following the configuration options provided in [23] [35]. The highest coherence value is  $\{0.4676\}$  found for 30 topics, iteration 1000 and the value of the hyper-parameters alpha 5 and beta 0.05. With these parameters, the value of  $K$  is decided as 30.

Generate the DevOps Topics. We found total 30 set of keywords from the SO dataset, each keyword set is candidate for a topic label. Each of the first two authors separately checked the 10 most popular keywords for each topic and checked at least 20 posts with highest correlation values with the topic. The other two authors reviewed the labels. Following previous work [36, 37, 38], we use the open card sorting method [36, 37] to mark the topic names. After agreement on the labeling, we merged a number of topics since their keywords had synonymous meanings. For example, two separate topics had top keyword set Topic15={git, repository, gitlab, jenkin, push, clone, remote, branch, commit, file} and Topic12={jenkin, build, branch, trigger, pipeline, git, gitlab, commit,

github, merge}. But they are essentially similar topics related with Git merging and building using Jenkins or GitLab, and so they were merged into a single topic labeled Git Build. In the end, we found 23 different topics for our dataset. After confirming the 23 topics, we revisited all topics to categorize into higher levels.

Tool names in topic labels. While labeling the topics and category names, we initially considered to give tool-agnostic name, e.g., instead of creating a sub-category of Jenkins, the tool-agnostic name could be Continuous-Integration. But after careful review of the top hundreds high correlation valued posts in those categories, it was evident that the topics did not discuss about generic continuous integration of other tools, rather the discussion was specific to the Jenkins tools and related pipeline or build processes. So, finally, we left the topic and sub-category names to be tool specific whenever was relevant. As mentioned in Section 7, there are other studies related with similar topic modeling of continuous software engineering processes which attempted to avoid tool specific topics, but ended up having tools names in topics and categories [39].

### 3.2.2. **Results**

We grouped the DevOps related questions into 23 unique topics and grouped the topics into four high-level categories: Cloud & CI/CD Tools, Infrastructure as Code, Container & Orchestration, and Quality Assurance. From Figure 4 shows the distribution of questions and topics under four major categories. Among the categories, Cloud & CI/CD Tool that has the highest percentage of questions (48.6%) and the highest number of topics (10 topics), followed by Infrastructure as Code (28.9% of the questions, 7 topics), Container & Orchestration (16.2% questions, 4 topics), and Quality Assurance (6.3% questions, 2 topics).

Figure 5 shows the 23 DevOps topics under four major categories ordered based on the distribution of questions (e.g., Cloud & CI/CD tools is the top and Quality Assurance is the bottom category). There are sub-categories under each major category (except Quality Assurance), e.g., Infrastructure as Code

has sub-categories of Automation and Security & Troubleshooting. The final topics are placed under those sub-categories, e.g., Automation sub-category have four topics. Now we discuss the DevOps topics under their categories.

- **Cloud & CI/CD Tools** In this category (48.6% questions) DevOps engineers discuss about deployment, continuous integration/continuous delivery (CI / CD), Jenkins workflow and build, test and debug project with Jenkins for on-premises and cloud project, etc. This category contains ten topics and these topics are grouped into four sub-categories: Jenkins, Azure, Git and IBM.

- (i) **Jenkins (21.4% questions)** This sub-category is divided into four topics areas: (1) *Jenkins Pipeline (13.3% question)* contains discussion on events or tasks that are linked in a sequence to do quick software release and create codes that automatically implement new software versions (e.g. ( $Q_{40454558}$ )). (2) *Build Mobile App (1.7%)* contains questions about continuous build, test and debugging of the mobile applications with Jenkins tools. (e.g. ( $Q_{19426477}$ )) (3) *Build Projects (3.4%)* contains discussion about end to end project building using Jenkins, build fail and error raised in Jenkins, etc. (e.g. ( $Q_{34130908}$ )). (4) *Distributed Architecture (3.0%)* discusses about installation and build process troubleshooting for distributed master slave nodes (e.g. ( $Q_{45845662}$ )).

- (ii) **Azure DevOps (12.5% questions)** sub-category contains discussions about Microsoft's DevOps and CI/CD tools. It covers three topics: (1) *Build Project (3.4%)* discusses about Azure DevOps automatic build, release, debug, troubleshooting questions (e.g. ( $Q_{62041467}$ )). (2) *DevOps Deployment (3.9%)* contains questions about Azure DevOps web/app service deployment, deployment configuration, and app deployment difficulties (e.g. ( $Q_{28638945}$ )). (3) *DevOps Pipeline (5.2%)* contains discussion about Azure DevOps project setup of an entire CI/CD pipeline (e.g. ( $Q_{58734429}$ ))

- (iii) **Git (11.7% questions)** sub-category contains two topics about the popular source code management tool and associated integration tool: (1) *Git build (8.2 % of the questions)* topic contains discussions about git configuration, build project code, GitHub project difficulties, tracking changes in the source

code, (e.g. ( $Q_{58473877}$ )) (2) *GitLab CI/CD* (3.5 % of the questions) contains questions regarding GitLab setup for build projects, CI/CD, and continuous testing as an integration tool (e.g. ( $Q_{57340112}$ )).

- (iv) **IBM Cloud (3.8% questions)** sub-category covers only one topic: (1) *Cloud Troubleshooting* (3.8%) contains questions about build, deployment, security management, in IBM's cloud platform for DevOps (e.g. ( $Q_{45775107}$ )).

• **Infrastructure as Code** This is the second highest category which covers 28.9% of the questions. It contains discussion about server configuration, cluster configuration and difficulties, file management, user access policy, infrastructure configuration scripts and troubleshooting, etc. This category has seven topics under two sub-categories as discussed below.

(i) **Automation (21.8% questions)** Sub-category contains four topics: (1) *Configuration Automation* (8.6%) discusses about on-premise infrastructure automatic configuration (i.e. make ansible-playbook, run the playbook for the given host etc. ( $Q_{45354448}$ )). (2) *Cloud Infrastructure Automation* (6.2%) contains questions regarding automatic cloud configuration including hardware, software, networking components, operating system (OS), and data storage components using different automation tools such as Terraform, Cloudformation etc. (3) *Script Syntax* (3.8%) contains common troubleshooting questions related with different scripting issues ( $Q_{28963585}$ ). (4) *Script Execution* (3.2%) is related with scripting, but the topic discusses more about infrastructure script execution procedures and challenges (e.g., permission settings, access controls of network devices or containers from certain scripts etc.).

(ii) **Security & Troubleshooting (7.13% questions)** contains discussion about security across the DevOps processes. This category contains three traditional IT infrastructural related topics whose titles are self-explanatory: File Management (2.7% posts, e.g.  $Q_{59468464}$ ), Exception Handling (1.9% posts, e.g.,  $Q_{53349416}$ ) and User Permission (2.5%, e.g.,  $Q_{39473843}$ ).

• **Container & Orchestration** contains 16.2% of DevOps posts that discuss about automating the deployment, scaling, networking, and management of

containers. This category comprises of two sub-categories.

(i) **Docker (2.7% questions)** only one topic is covered by this sub-category:

(1) *Docker Manager (2.7%)* provides the ability to write code in one version and ship it to any platform, it provides the complete containerization.

(ii) **Kubernetes (13.4% questions)** is an open-source system for orchestration of containerized applications. There are three topics under this sub-category: (1) *Kubernetes Cluster (5.0%)* is a collection of nodes that run containerized applications. (2) *Kubernetes POD (4.8%)* discusses about Kubernetes cluster control management, Kubernetes cluster metric monitoring, operation (e.g. ([Q<sub>52066896</sub>](#))). (3) *Traffic Management (3.6%)* contains discussion about resilience in Kubernetes with advanced traffic management, traffic routing, traffic shaping, and traffic monitoring (e.g. ([Q<sub>60964361</sub>](#))).

• **Quality Assurance** contains 6.3% of the questions. This category describes various aspects of testing in DevOps environment including automatic testing configuration, performance testing, unit and other testing issues. It contains two topics:

(1) *Functional Test Automation (3.2%)* topic contains posts on continuous integration testing, unit testing configuration, acceptance testing, setting up automatic functional testing and issues (e.g., Running iOS project unit tests from Jenkins does not produce any output ([Q<sub>10751086</sub>](#)), dotMemory Unit Standalone launcher with XUnit hang at the end of the test ([Q<sub>53535857</sub>](#))). (2) *Non-functional Test Automation (3.1%)* contains discussion about load test, performance test, Disaster Recovery, Security, Accessibility, Usability and Operational testing (e.g., How to publish Rough Auditing Tool for Security (RATS) xml results in Jenkins ([Q<sub>14585222</sub>](#)))

**RQ<sub>1</sub>** What types of topics are discussed about DevOps in SO? We found 23 topics that are divided into four categories: Cloud & CI/CD Tool, Infrastructure as Code, Container & Orchestration, and Quality Assurance. Cloud & CI/CD Tools category covers nearly 49% questions. Jenkins Pipeline under Cloud & CI/CD Tools category is the most discussed topic with 13.3% of total posts, followed by Configuration Automation topic (8.6% posts) under Infrastructure as Code category.

### 3.3. RQ<sub>2</sub> DevOps Topic Popularity vs Difficulty

#### 3.3.1. Approach

We compute three metrics to measure popularity of each topic: 1. Average number of views for all posts of a topic, 2. Average number of questions of a topic marked as users' favorite, and 3. Average score of questions of a topic. We determine the difficulty of the topic using two metrics of the question, 1. Percentage of the questions without any accepted answer, and 2. Average time needed to receive an accepted answer. These mentioned five metrics were also previously used in several research papers to compute the popularity and difficulty of topics found in SO posts [40] [23] [41]. Finally, we also assess the correlation between each of three topic popularity and two difficulty metrics. We use Kendall's  $\tau$  correlation measure [42]. Unlike Mann-Whitney correlation [43], Kendall's  $\tau$  is not susceptible to outliers in the data.

#### 3.3.2. Results

**Topic Popularity**, Table 3 shows three popularity metrics for each DevOps topic: average number of 1. view count 2. favorite count 3. score. Topics are ranked in descending order by average view count under each category.

File Management under Infrastructure as Code category has the highest number of average view count (2,805). This topic is comprised of different file management issues which can vary from general purpose file to various tool specific files such as Dockerfile, build log file, timestamped war file e.g. Q<sub>43161244</sub>. Second most viewed topic is Jenkins Pipeline under Cloud & CI/CD

Table 3: DevOps topics, their popularity, and difficulty

Topic	Popularity score			Difficulty score		
	Avg view	Avg fav.	Avg score	W/O acc. ans.	Med. acc.	Hrs
	<b>Cloud &amp; CI/CD Tool</b>					
Jenkins Pipeline	2673	0.53	2.19	61.7%	194	
Jenkins Distributed Architecture	2332	0.52	1.90	64.7%	174	
Git build	2402	0.71	2.52	60.4%	229	
Jenkins Build Projects	2035	0.46	1.78	63.3%	154	
Jenkins Build Mobile App	1543	0.74	2.38	61.1%	248	
GitLab	1416	0.57	2.09	60.1%	250	
Azure Build Project	1391	0.46	1.93	44.4%	164	
Azure DevOps Deployment	933	0.69	1.75	56.8%	142	
Azure DevOps Pipeline	607	0.24	1.09	45.8%	132	
IBM Cloud Troubleshooting	421	0.16	0.66	59.5%	74	
<b>Infrastructure as Code</b>						
File Management	2805	0.40	2.05	58.1%	191	
Script Execution	2603	0.39	1.75	59.1%	135	
User Permission	2509	0.63	2.23	58.6%	216	
Configuration Automation	2125	0.45	1.87	51.6%	145	
Exception Handling	1970	0.42	1.74	65.6%	132	
Script Syntax	2018	0.29	1.44	44.9%	73	
Cloud Infrastructure Automation	1054	0.38	1.62	57.4%	133	
<b>Quality Assurance</b>						
Non-func Test Automation	2062	0.46	1.95	63.7%	251	
Func Test Automation	1400	0.50	1.65	63.9%	197	
<b>Container and Orchestration</b>						
Docker Manager	1435	0.60	1.94	59.1%	138	
Kubernetes POD	1381	0.46	1.89	57.2%	127	
Kubernetes Cluster	1020	0.42	1.44	59.9%	106	
Traffic Management	922	0.38	1.38	61.8%	107	
<b>Average</b>	<b>1695</b>	<b>0.47</b>	<b>1.8</b>	<b>58%</b>	<b>161.4</b>	

Tools category (2,673 views). This is a merged topic comprising of scripted and descriptive Jenkins pipeline, overall Jenkins job flow, and environment variable settings. Example posts are  $Q_{56887022}$ ,  $Q_{46429657}$ .

The topics Script Execution (view 2,603) and User Permission (view 2,509) from Infrastructure as Code category are the third and fourth most viewed topics respectively. Both of these topics are related with general infrastructure problems related with any DevOps tools e.g., “How to call batch/groovy script located in jenkins slave from “Execute system Groovy script”?” ( $Q_{15172048}$ ), “How to make Windows START command accept standard input via pipe and pass it on to the command it invokes?” ( $Q_{34512012}$ ). Git Build has the highest average favorite posts (2.52) followed by Build Mobile App (favorite score 2.38). IBM Cloud troubleshooting is the least viewed topic (421 views) followed by Azure DevOps Pipeline (607 views).

**Topic Difficulty.** Table 3 contains the difficulty metrics per topic 1. Percentages of question without any accepted answer, and 2. Median hours taken to receive an accepted answer.

Among all the topics, Exception handling from infrastructure as Code category is ranked the most difficult topic because around 65.6% of its questions remain without accepted answers, while those that get an accepted answer normally had to wait a median of 132 hours. The following most difficult topic is Jenkins Distributed Architecture whose unaccepted post rate is 64.7% with a median wait time of 174 hours for accepted answers. Non-functional Test Automation topic has the highest median waiting time of 251 hours for accepted answer. A non-functional test automation question asked 4 years ago still got no accepted answer  $Q_{44236838}$ , whereas another question with 20K views ( $Q_{18779450}$ ) from the same topic got its accepted answer after four years of asking. Questions from GitLab topic have median waiting time for 250 hours for accepted answer. For example, a question  $Q_{31687499}$  was asked more than 6 years ago, still did not get any accepted answer.

**Correlation between Topic Popularity and Difficulty.** Our results from

Table 4: Correlation between topic popularity &amp; difficulty

Coefficient/p-value	<b>View</b>	<b>Favorites</b>	<b>Score</b>
<b>% w/o acc. ans.</b>	0.138/0.373	0.173/0.254	0.123/0.413
<b>Med hrs to acc. ans.</b>	0.375/0.012	0.566/0.0002	0.655/1e-05

topic popularity and difficulty does not indicate any positive or inverse relationship between all the popularity and difficulty metrics of a topic. For example, the most popular (viewed) topic File Management has 58% unsolved questions. Similarly, the least popular topic of Cloud Troubleshooting-IBM also has 59% unsolved questions. On the contrary, the most difficult topic Exception Handling (65.6% unaccepted questions) has an average view of 1970 and favorite score 0.42, whereas Configuration Automation topic has similar view and favorite score (2125 view and 0.45 score), but it is the fourth easiest topic with 51.6% unaccepted questions. Popularity does not correlate with solution of questions.

Similar statistics can be found from Table 4 which shows six correlation measures (using Kendall’s  $\tau$ ) between topic difficulty and popularity in Table 3. All six correlation coefficients are positive and three of them (related with median hours to acceptable answers and popularity metrics) are statistically significant with more than 95% confidence level. From this correlation, it shows that more popular topics can take more time to get an acceptable answer. But there is no significant correlation between popularity and percentage of questions without accepted answer.

**RQ<sub>2</sub> How do the popularity and difficulty of DevOps topics vary?**

File Management topic, from the infrastructure as Code category, is the most popular in terms of post views. Exception Handling, from the same category, is the most difficult in terms of getting an accepted answer.

Non-functional Test Automation takes the longest 251 hours to get an acceptable answer. There was no statistically significant inverse correlation between popularity and difficulty of topics.

#### 4. Survey of DevOps Practitioners

In this section, we report a survey of DevOps professionals that we conducted to validate our empirical study findings and to get additional insights about DevOps challenges, if there are any. By analyzing the survey responses, we answer two research questions:

**RQ3.** How the industry practitioners agree with our empirical study findings?

**RQ4.** What additional insights about DevOps challenges can we get from open-ended responses of survey participants?

**Survey Participants.** To conduct the survey, we targeted industry professionals in small, medium, and large corporations. To select the participants, we combined judgemental sampling [44] with snowball sampling [45]. In judgemental sampling, we start with participants whom we knew had already experience in DevOps as architect, manager, developer, or operational engineer.

Around 55% of participants had more than 10 years of professional experience, and 95% had at least two years experience. Hence, the participants in general had sufficient hands on professional experience. They had diversified roles in their organizations: 35% solution architects and 15% managers (delivery or operations), 30% were development engineers, and 15% were operations engineers. Around 48% of participants came from small organizations (1-50 employees), 32% from small to medium organizations (51-150 employees), and the

rest (20%) from medium to large organizations (150+ employees). 45% of participants evaluated their organization to be in the mid-level in terms of DevOps maturity and 50% consider they have advanced implementation and practice of DevOps. Distribution of both the size and maturity of the organizations are not skewed and can be considered well-balanced representation of general organizations. The survey questions and anonymized responses are shared in our replication package [14].

**Survey Questions.** All survey questions are listed in Table 5 and 6. Table 5 contains the fixed answer questions (multiple choice and checkboxes) and Table 6 contains open-ended questions. The detailed motivation of each question is discussed in the following subsections along with the results and analysis.

Table 5: Survey questions with fixed answers. Target research questions are denoted in the RQ column.

#	RQ	Survey Question	Ans. Type
Q1	Info	What is the size of your engineering organization?	Multiple Choice
Q2	Info	On a scale of 1 to 5 where your team or organization positions in practice of DevOps ?	Multiple Choice
Q3	Info	What is the earliest year your organization started devOps practices?	Multiple Choice
Q5	RQ1	What tools do you use as part of your devOps practice? (You can write additional tools in other box as comma separated)	Checkboxes
Q6	RQ4	What do you think the most critical part of devOps life cycle?	Multiple Choice
Q8	RQ4	When a new project is ready to go live, which devOps part you initiate first?	Multiple Choice
Q9	RQ2	Please choose difficulty level for each of the tools/topics below:  [Build (Ant, Maven, Gradle)] [Quality Assurance (SonarQube, Selenium, JMeter, ...)] [Integration Pipeline (Jenkins, ...)] [Configuration Automation (Ansible, Chef, Puppet, ...)] [Containerization & Orchestration (Docker, DockerSwarm, Kubernetes, ...)] [Cloud Infra Automation (Terraform, CloudFormation, ...)] [Monitoring (Splunk, ELK, Grafana/Prometheus), ...]	Multiple Choice Grid
Q12	RQ2	How often your developers face problems related with basic IT Infra (e.g., repository or file settings, user permission settings, scripting syntax or execution permission etc.)	Multiple Choice
Q13	RQ2	How often you get solutions from community or Stack Overflow about the devOps problems you face?	Multiple Choice
Q14	RQ1	Which cloud services do you use	Checkboxes

Table 6: Survey Questions with open ended answers. Most of these were related with RQ4 (industry insights) and some questions also helped analyze RQ1 as denoted under column RQ

#	RQ	Question	Type
Q4	RQ4	What organizational process/training you follow to adapt devOps practices and tools?	Open-ended answer
Q7	RQ4+RQ1	What difficulties do you face in this critical phase of devOps that you work on?	Open-ended answer
Q10	RQ4+RQ1	What are the advantages of the continuous integration tool you primarily use (e.g. Jenkins or Gitlab CI etc.)?	Open-ended answer
Q11	RQ4+RQ1	What challenges you face while using your primary integration tool (you may also mention status of community support)?	Open-ended answer
Q15	RQ4	Do you think Cloud infra automation (Terraform, CloudFormation etc.) will be more used in future? If so, why? If not, why?	Open-ended answer

#### *4.1. RQ<sub>3</sub> Validation of Empirical Study Findings*

##### *4.1.1. Approach*

We designed the survey questions around the key findings of our first two research questions as follows.

**Questions regarding RQ1.** We attempted to match the concepts found in DevOps literature such as continuous software engineering processes (e.g., CI/CD, continuous testing etc.) with the industrial practice of the professionals - which processes they start first in DevOps initiative, and what are the crucial processes and their challenges in DevOps. Moving forward from the DevOps concepts, there were questions about the popular tools - the list, category, and usage of tools by the practitioners. We listed and grouped the tools based on the findings of the empirical study. In every case, we also added option to provide new tools or categories.

**Questions regarding RQ2.** As we found from the empirical study that continuous integration tools are one of the most discussed topic, we asked about advantages and challenges of the integration tools using open ended questions.

##### *4.1.2. Results*

In Table 7, we validate key findings from RQ1 and RQ2 against the results of RQ3 from the survey. We group the findings by the two RQs. For example, for RQ1, we have three consolidated key findings in Table 7 (E1-E3), and we find three key insights (S1-S3) from our survey responses regarding E1-E3; and there are two findings compared from RQ2 (E4, E5). The left column in Table 7 shows the key findings from the empirical study and the middle column shows the corresponding validated insights from the survey responses. For every finding pair, we provide our remark on the right column.

Among the overwhelming number of tools and posts regarding DevOps, in our quantitative study one tool particularly stands out which is Jenkins. Despite being the most popular tool, 61.1% to 64.7% of Jenkins related posts are unsolved in SO, and all four topics of Jenkins are included among the top ten most difficult topics of DevOps. With couple of open-ended questions (Q10,

Table 7: Mapping of key findings between the empirical study and the survey responses

Empirical Study Finding (E)	Survey Finding (S)	Remarks (R)
<b>RQ1. What types of topics are discussed about DevOps in SO?</b>		
E1. 64.8% of questions in DevOps are related with specific tools (under Cloud & CI/CD and Container & Orchestration categories) and total 14 out of 23 i.e. 61% topics were tool specific. Automation (with tools) is the most discussed sub-category (21.8% of all DevOps questions).	S1 (Q5, Q7, Q14). 20 respondents listed total 39 tools and 9 cloud services they use related with DevOps. One participant responded <i>"Too many tools and practices. Each team must explore do trial and error and settle on what best works for the team."</i>	R1. There are plethora of useful tools for practicing DevOps, and proper usage of tools is a critical part for successful DevOps implementation.
E2. The CI/CD engine Jenkins (21.4% questions), the second most discussed sub-category, is the most discussed tool. Another CI/CD tool GitLab CI/CD had 3.5% questions (one sixth of Jenkins) in SO.	S2(Q5) Jenkins is used by 65%, GitLab CI/CD is used by 35%, and Circle CI/CD is used by 5% of respondents.	R2. Jenkins usage is 2 times, but Jenkins related questions in SO are 6 times of GitLab CI/CD. This could either mean GitLab CI/CD is a simple stable tool or Jenkins is more difficult to use tool.
E3. Cloud platform topics: 12.5% posts on Azure DevOps, 4.6% Amazon CloudFormation 3.8% IBM Cloud.	S3 (Q14) Usage: Azure used by 30%, Amazon CloudFormation used by 10% participants. No participants use IBM Cloud.	R4. Azure is used 3 times and also discussed in SO around 3 times of CloudFormation. The empirical and survey data seems to be congruent.
<b>RQ2. How do the popularity and difficulty of DevOps topics vary?</b>		
E4. Jenkins is the most popular tool in terms of number of posts in SO. But Jenkins topics require higher range of waiting period to get acceptable answers (from 154 to 248 hours in average, whereas median waiting period is 150 hours).	S4 (Q10, Q11). Most respondents mentioned Jenkins is easy to use. They also do not consider long answer time in SO to be a major problem.	R4. From empirical study data, Jenkins appeared to be a more difficult topic (high unsolved questions and waiting time), but from practitioners' response, Jenkins rather seems to be a useful, easy to use tool.
E5. Out of 174K DevOps posts in Stack Overflow, 101K posts (58%) do not have solution and it takes 161.4 hours to get an accepted answer.	S5 (Q13). 68.4% respondents perceive that Stack Overflow provides solutions <i>most cases</i> and 10.5% feels that they get solutions <i>every time</i> .	R5. The respondents seems to be getting more support from Stack Overflow than the empirical data presents.

Q11), we wanted to validate whether practitioners also feel the similar difficulties in Jenkins. The topic analysis from those open-ended questions are shown in Figure 8, and Figure 9.

Slightly contrary to the research finding, the participants seemed to consider Jenkins a user-friendly helpful tool. The overall impression can be reflected by a response from a participant *“It [Jenkins] is user-friendly, easy to install and does not require additional installations or components. It is free of cost. Easily Configurable. Jenkins can be easily modified and extended. It deploys code instantly, generates test reports. Platform Independent. Jenkins is available for all platforms and different operating systems, whether OS X, Windows or Linux. Rich Plugin ecosystem. The extensive pool of plugins makes Jenkins flexible and allows building, deploying and automating across various platforms.”*

It is possible that because of the ubiquity of Jenkins, a lot of unsolved/unanswered questions are trivial or duplicate and not responded by the active community in SO. Consequently, we can conclude the ease of Jenkins, but we cannot conclude the reason of difference between the empirical study and survey.

**RQ<sub>3</sub> How the industry practitioners agree with our empirical study findings?** The survey responses mostly align with the findings of RQ1 and RQ2. 61% DevOps topics were specific tool related, and the participants also reported using 39 tools and 9 cloud services for DevOps. In terms of difficulty of the topics in SO, respondents seem to be more positive of solutions provided by SO. In SO analysis, we found Jenkins has 6 times posts of GitLab, but in survey, Jenkins was used 2 times of GitLab.

#### 4.2. **RQ<sub>4</sub> More Insights about DevOps Challenges**

##### 4.2.1. **Approach**

While most of the survey questions were around the validation of the empirical findings, we also added five open-ended questions for qualitative analysis on challenges of DevOps processes (CSE topics), tools (CI/CD tools), and on future trend (cloud infrastructure automation) as listed in Table 6. To analyze

the responses, we applied an open coding approach [46] that includes labeling of concerns/categories in textual contents. In our open coding, we followed the card sorting approach [37]. For example, the following response “it’s error free and suitable for DR” has two different conceptual coherent quotes, “Error Free”, and “Suitable for DR”. The first quote refers to quality assurance, and second one ‘disaster recovery’ (DR) refers to automation of infrastructure as code in DevOps context. The first two authors coded the open-ended survey responses to extract potential categories independently. The level of agreement between the coders was calculated using Cohen’s  $\kappa$  [47]. The value of  $\kappa$  was 0.63 referring to a substantial agreement.

#### 4.2.2. Results

Table 8: The topic or concern categories observed during the open-coding of the responses (#TC = Total Categories, #TP = Total number of participants, #C = the number of code, #P = the number of participants referring the code)

Topic or Concern	Q04		Q07		Q10		Q11		Q15		#TC	#TP
	#C	#P										
Agile Development	4	4					2	2	1	1	7	7
Cloud & CI/CD Tools	3	3	3	3	15	8	8	8	7	6	36	28
Community Support							5	3			5	3
Container & Orch.	6	4	1	1					1	1	8	6
Continuous SW Engg.	7	3	5	4	11	6	9	8	5	1	37	22
Cost Management					2	2			1	1	3	3
Infrastructure as Code	3	3	9	6	15	11	5	5	19	11	51	36
Quality Assurance	1	1							2	2	3	3
Skill Development	11	6	4	4			2	2	2	1	19	13
Version Compatibility							6	3			6	3
Total	35	24	22	18	43	27	37	31	38	24	175	124
# Categories/Question	7		5		4		7		8			

The open coding of those survey question responses resulted in a total of ten topic or concern categories. Figure 6 lists hierarchy and count of each

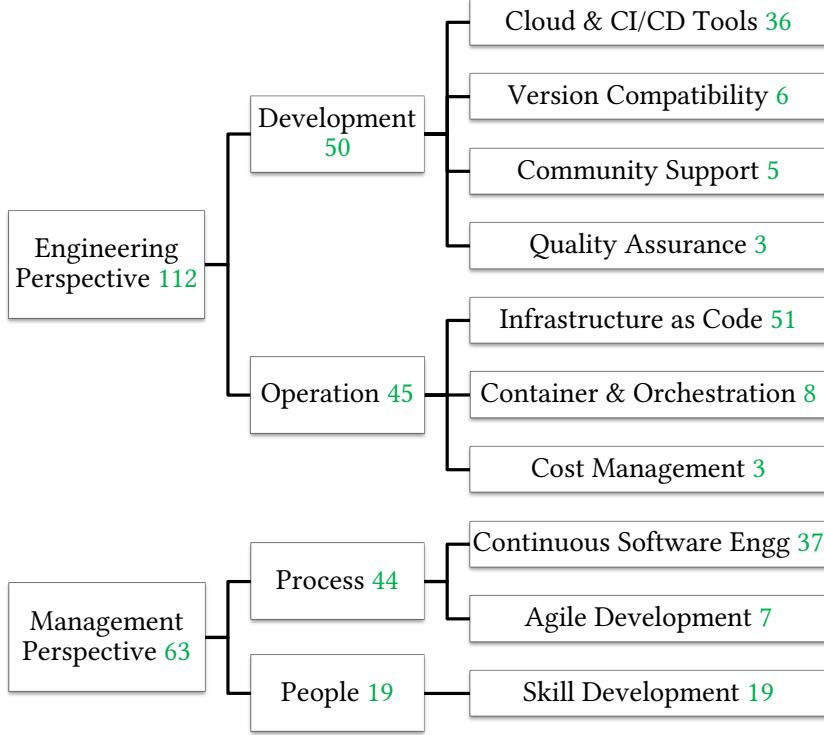


Figure 6: Hierarchy of topics observed in qualitative survey

topic found in the survey. We used conceptual map of [13] to categorize the topics. In Table 8, we show the number of quotes for each topic as we found in our survey data. In following subsections, we explain the categories by each open-ended question (Q4 regarding organization DevOps training, Q7 about DevOps processes, Q10 about CI/CD tool advantages, Q11 regarding challenges of CI/CD tool, and Q15 about the opinion of cloud automation’s future).

**Importance of Continuous DevOps Processes (Q6, Q8):** As noted in Background Section 7, there are important continuous software engineering processes in the DevOps concept. The empirical study did not directly shed light on those processes. So we included few open ended questions in the survey to get industry feedback. 45% of the respondents view continuous integration as most critical part of DevOps, whereas 25% emphasized on continuous monitor-

ing as most critical process. Others almost equally view continuous deployment, testing, and building.

In a slightly different perspective, when asked which DevOps practices they start in any new project, the respondents were almost equally divided among all major processes of Continuous Testing (15%), Containerization & Orchestration (15%), Configuration Automation (15%), Complete Integration Pipeline (25%). Here is an excerpt from a participant about the importance and risks of deployment as “*So many things can go wrong during deployment. Deactivating the currently running version while deploying the new one is complicated with full ecosystem support. Without properly testing the deployment during staging, migration can go wrong. For large applications in Java, it still needs manual intervention.*”

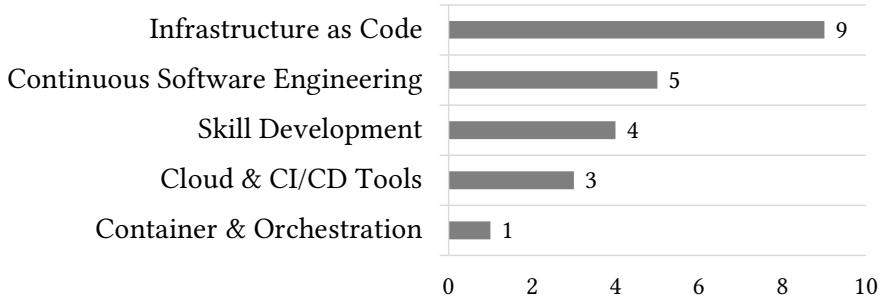


Figure 7: Continuous Software Engineering Topics

**Difficulties of DevOps Processes (Q7):** 30% respondents mentioned the difficulties to continuous integration and testing among multiple cloud services (e.g., between AWS and Azure), and adaptation of various dependent and connected systems/tools is a major challenge. 30% of practitioners think that keeping engineers aligned, and skilled with DevOps practices is challenging. The topics responded by participants is listed in Figure 7.

**Difficulty Level of Process Specific Tools (Q9):** Since 60% of DevOps topics were tool specific, instead of asking the difficulty level of the 23 general topics, we asked the practitioner their opinion on the tools they use. As also

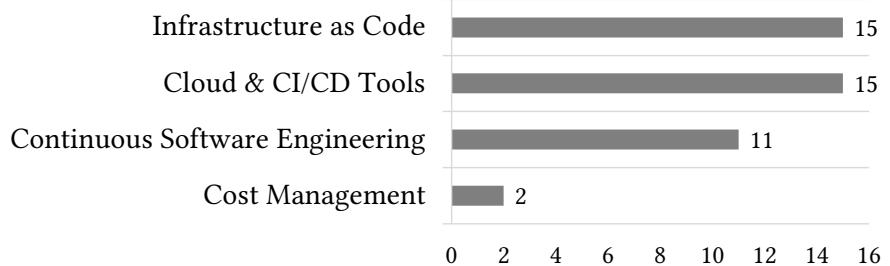


Figure 8: Responded topics in advantages of integration tool

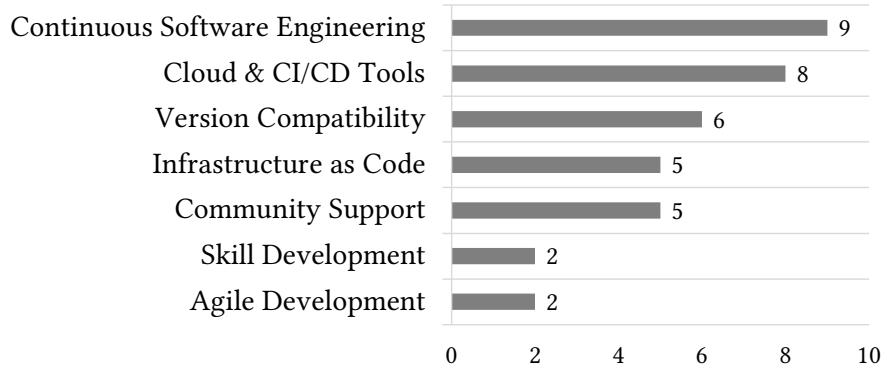


Figure 9: Responded topics in challenges of integration tool

noted that there are numerous DevOps tools, we categorized the tools according to the DevOps processes and collected the industry feedback. Build tools found to be easiest (by 42% respondents), while Containerization and Monitoring tools seemed to be most difficult (by 11% participants), and Cloud infra automation tools were least used (53% respondents did not use any such tool). The details are provided in the Table 9.

**Skill Development in DevOps (Q4):** We wanted to know what organization are doing for training and skill development in DevOps area. 20% practitioners think that lack of skill is the most common problem in DevOps. But only 5% companies have formal training in DevOps. 10% participants learn DevOps from online courses such as Udemy, Plural Sigh. One respondent mentioned

Table 9: Difficulty level of different tools set by DevOps continuous processes. Each colored bar denotes a difficulty level. Black = Not used, Green = Easy to use and learn, Magenta = Easy to use after some experience, Orange = Frequent troubleshooting needed, Red = Even SO cannot answer

Process wise tools	Difficulty level				
Build Tools	 42	 53	 5		
Integration and Pipeline Tools	 26	 58	 11	 5	
Containerization Tools	 53	 32	 11	 5	
Monitoring Tools	 5	 42	 26	 11	 16
Quality Assurance Tools	 21	 47	 11	 5	 16
Configuration Automation Tools	 5	 32	 21	 5	 37
Cloud Infra Automation Tools	 32	 5	 11	 53	

*“most of the knowledge transfer is done ad-hoc by the mentor or other senior engineers.”*. Topics referred in the question is shown in Figure 10.

**Future Trend of Cloud Infra. (Q15):** The participants opined that cloud infrastructure automation will be a critical part of DevOps and programmers will have to improve their infrastructural skills to keep up. Figure 11 shows responded topics.

**RQ<sub>4</sub> What additional insights about DevOps challenges can we get from open-ended responses of survey participants?** 45% of participants consider continuous integration as the most critical component of DevOps practice. Though, none of 23 topics found in RQ<sub>1</sub> refers to monitoring tools, 25% participants considered continuous monitoring to be the most important DevOps process. The respondents emphasized lack of skilled professionals as a major problem and only 5% organizations offer formal training.

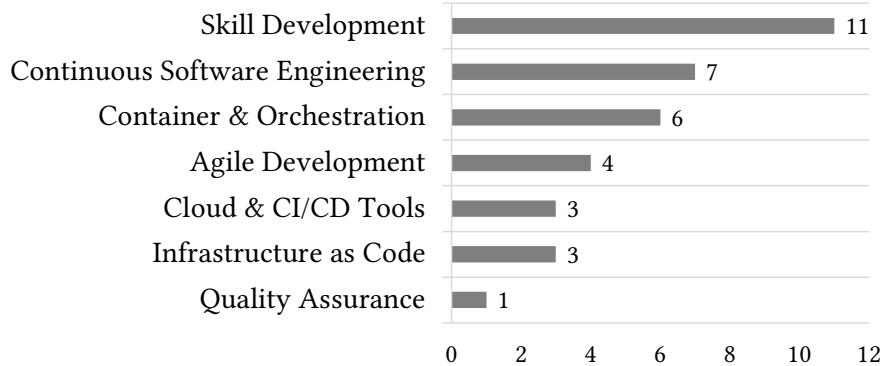


Figure 10: Skill Development Requirement Topics

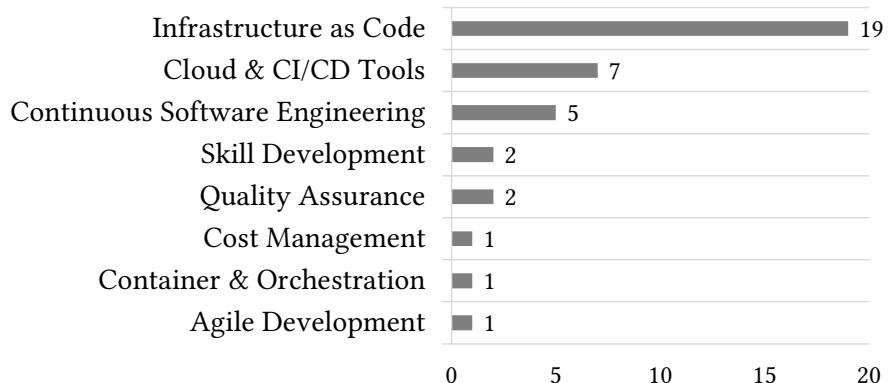


Figure 11: Responded topics for Cloud Infra trend

## 5. Implications of Findings

Our study findings can guide the following DevOps stakeholders: 1. DevOps Vendors to develop more usable DevOps tools, 2. Organizations to allocate more resources to train the DevOps practitioners, 3. Solution Architect to decide on tools like Kubernetes for container and orchestration, 4. Delivery/Operations Manager to use the popular automation tools and to collaborate with solution architects to devise fallback and recovery plan, 5. DevOps Practitioners to determine learning paths based on DevOps topics, and 6. DevOps Researchers and Educators to develop innovative DevOps technology to increase the rapid

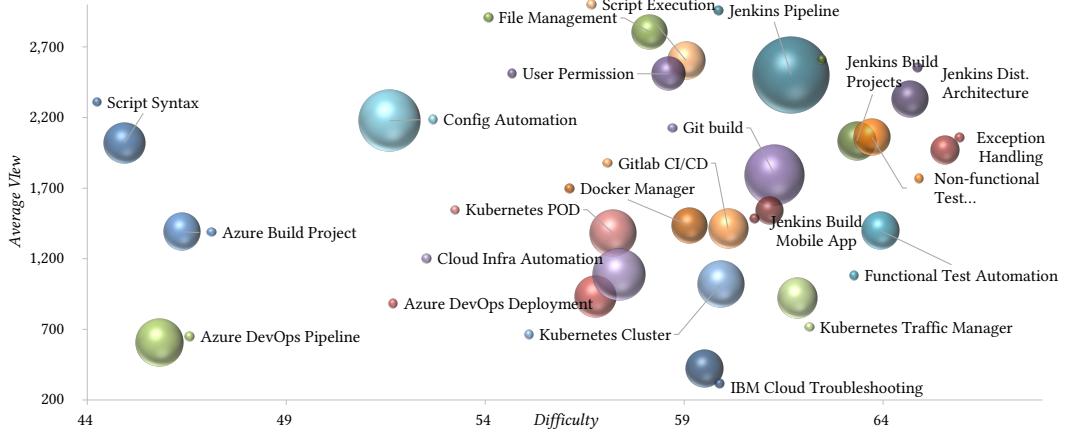


Figure 12: DevOps Topics popularity vs. difficulty

adoption and usage of DevOps tools. We discuss the implications below.

**DevOps Vendors.** In Figure 12, we show a trade-off between the popularity and difficulty of the 23 DevOps tools and topics in our dataset. The more right a topic is the more difficult it is. The higher a topic is at the top, the more popular it is. The size of each bubble represents the total number of questions. Jenkins Pipeline is one of the most difficult topics and also among the most popular ones. Therefore, Jenkins vendors need to make the pipeline creation process and techniques in Jenkins easier to use. Similarly Cloud CI/CD developers can also get insight as indicated by experts in the qualitative survey *“To make the data centres and systems entirely configurable is going to be the demand for the next generation devOps engineers where they will make all the cloud infrastructure configuration settings possibly only by the code they will write”*.

**Software Organizations.** While inquiring about the difficulty of DevOps topics to our survey participants, we find that very few (5%) of them have formal support for training from their companies. This is irrespective of our survey participants coming from small, medium and large organizations. Therefore, software organizations can allocate more training for DevOps practitioners. Formal

and frequent training are important, given many practitioners may not always find solutions to their problems in forums like SO (having 58% unsolved questions).

For conducting DevOps related training, one important part is preparing the syllabus of training. Our paper provides a complete mapping of all DevOps related concepts and hence, can help a lot as a training outline. Moreover, in all aspect of DevOps, we also listed down most notable tools under section 3. As we found from the practitioners' survey that tools are very important part of DevOps, organizations can also run such tool specific training.

**Solution Architects.** From our study, active communities in DevOps topics can be found and architects can consider community support factor for selecting technologies. For example, three topics of Kubernetes (pod, cluster, traffic management) have average difficulty level of around 60% which is just above average difficulty level (59%) of all topics (see Figure 12). We find that these topics have one of the fastest accepted answers from the community which is around 106 to 127 hours on average (among top 5 topics with fastest resolution). Because of this active community support, solution architects can more confidently use Kubernetes in their solution design. On the other hand, comparing between Jenkins and GitLab for continuous integration, a team may reconsider choosing GitLab which has slowest community support (250 hours of waiting period for acceptable answers)

Moreover, our study can be used as an index for tools and techniques by solution designers. Solution architects who are designing a new system can identify important DevOps practices from Figure 5 for building an end-to-end automated pipeline and can find detail tools in section 3. For example, they may be already aware of configuration automation (using Ansible, Puppet etc.), but they can also instantly learn about cloud infrastructure automation and related tools (terraform, cloud formation etc.) which are an emerging technology.

**Delivery/Operation Managers.** The study can be used for planning DevOps adoption process in organizations. For example, in the Figure 12, Configura-

tion Automation is represented by the second biggest bubble as its post count (16,000) is more than 2.5 times of the median post count (6,000). The posts are also frequently resolved, as the bubble lies at the left of the chart. Using insight of this quantitative result, managers who want to adopt or initiate DevOps practices can start by automating configurations with related popular tools such as Ansible, Chef, Puppet, etc.

Our study can also help managers in making realistic delivery plans. Apparently, it may seem that using DevOps tools will automate the delivery cycle and ensure faster production rollouts. However, our study shows that setting up DevOps tools are not always easy and developers often struggle with a lot of difficulties throughout the pipeline.

For example, if a team is suddenly stuck with mobile application build through Jenkins, our paper should be able to guide a manager that this is a very difficult situation to handle, since almost 61.1% SO posts does not have any resolutions and even if they get lucky, the median wait time can be as long as 10 days (248 hours to be precise), as listed in Table 3. Hence, managers studying our paper should be able to anticipate the roadblocks and can help their team with proper planning and necessary resources.

**Developers.** Our qualitative analysis has identified the need for Skill Development in DevOps. A respondent specifically emphasized on the infrastructure skill of developers by saying “*As it [cloud infra automation] is mostly used by people having programming exposure, the lack of system and networking knowledge in this set of people is a major problem in my opinion.*”. This also validates the quantitative finding on Infrastructure as Code category. Topics such as file & permission management, script execution have less number of posts, but their average view count is highest among all topics (top side in the Figure 12). Therefore, aspiring developers can start learning tools related to these topics and then move to more advanced topics like Jenkins pipeline.

In this paper, we have reported that most knowledge transfer regarding DevOps happens ad-hoc without any formal training. It is only to the best

interest of the developers that they learn about the concepts and tools online by themselves, since we have identified that only 10% of developers take online DevOps courses to improve their skill.

**Educators and Researchers** can create tutorials and documentation for the most difficult topics as shown in Figure 12. Researchers can also investigate the reason of delay in getting acceptable solutions in SO posts. In addition, DevOps research could take cues from crowd-source and big-data research to develop techniques that can automatically find acceptable answers to unanswered questions, e.g., by recommending acceptable answers to a question [48, 49].

## 6. Threats to Validity

**External Validity** threats concern the generalizability of our findings. We focused on SO, which is one of the largest and most popular developers' Q&A Web sites. Yet, our findings may not generalize to other Q&A Web sites. We only considered questions and accepted answers in our topic modeling. Our approach is consistent with previous work that used topic modeling on SO data [50, 51, 52, 53, 48, 54, 55, 56].

**Internal Validity** threats concern experimental bias and errors while conducting the analysis. In particular, in our study, we manually labeled the topics. To reduce any bias in this labeling, two different authors separately labeled the topics and then another author, who is a domain expert, validated the labeling. The three authors discussed any conflict and resolved them via discussions. Thus, we believe that we reduced labeling bias to an acceptable minimum.

**Construct Validity** threats relate to potential errors that may occur when extracting data about DevOps-related discussions. We collected all SO posts labeled with one or more tags related to DevOps, i.e., 19 different tags. We created the list of tags using state-of-the-art approaches [57, 55] and by manually verifying the tags. Threats to construct validity also pertain to the difference between theory, observation, and results. Our use of metrics to measure popularity and difficulty fall under such threats. Yet, we used metrics that were

used in previous works [41, 23], thus mitigating the risk of wrong measurements. In the questionnaire survey, asking people to indicate a context free difficulty level for a broad category of topics may introduce some inaccuracy to measure these perceptions, since difficulty level of a task varies from technology to technology in some cases. However, the participants are likely to consider majority applications while responding to such queries and some peripheral cases may be overlooked in absence of better alternative method.

## 7. Related Work

**DevOps Definition.** Aiello et al. defined DevOps in [15] as *DevOps is a set of principles and practices intended to help development and operations collaborate and communicate more effectively*. In a literature survey on DevOps [13], Leite et al. identified four major categories of DevOps concepts, namely, Process, People, Delivery, and Runtime using a systematic analysis on 50 'core' DevOps publications till 2019. While process and people are relevant from the management perspective, delivery (development related) and runtime (operation related) are associated from the engineering perspective.

**Case and Industry Studies on DevOps.** Besides the academic researches, there has been a industry wide survey and state of DevOps report published by Puppet since 2011 [58]. They surveyed 2,400 industry-people in 2020 [59] and in 2021, the report investigates team interactions and recommends actions toward “breaking down the middle”. There are few more studies on DevOps definition, literature survey, and case studies on DevOps practices and adoption [60, 61, 62, 4, 63, 64, 65] and as well as a literature survey on DevOps along with qualitative interview with six organizations on their practices [66]

**SE Research on DevOps.** There are few research works that focused on different aspects of DevOps practice. The studies on Continuous Software Engineering [39], Release Engineering [12], Dockerfiles [67] and Docker [68] provide insight on the specific processes or tools of DevOps engineering. The research [68] presents a study of LDA-based topic modeling on the community discuss-

sion related only with Docker on topics. The work of [67] concentrates on the quality and evolution of Dockerfiles.

**Research on Continuous Software Engineering (CSE).** A closely relevant work by [39] performs empirical study on CSE. As outlined by the authors of [69, 16, 70] and also in the background section, CSE has few common processes (continuous planning, continuous testing, CI/CD etc.) as DevOps, but DevOps is a much broader concept covering organizational change management and engineering practices [13].

In the empirical study on CSE, the authors also explicitly attempted to avoid Stack Overflow posts concerning the tools of DevOps, still 12 out of 32 topics, identified by LDA-based topic modeling, were directly related with specific DevOps tool. At the same time, 186 out of 200 posts qualitatively analyzed by the study were directly related with DevOps tools. Despite the researchers' attempt to avoid DevOps tool, their inevitable presence in topics and posts refers that tools are a significant part of DevOps. Though our study apparently overlaps with that research, our empirical study comprises of 13 times larger dataset and 23 consolidated topics focused all around DevOps. Moreover, we conducted an external practitioner survey to validate the empirical study and collected qualitative insights on DevOps on people and process concepts.

The major technical topics missing in the CSE study is automation. The paper did not report any topic regarding automated testing, configuration automation or cloud infrastructure automation which comprises 18% of DevOps post we found. Secondly, their paper missed leading configuration management tools such as Ansible, Puppet and major cloud infrastructure automation tool Cloud Formation. The study did not mention Kubernetes and did not discuss any topic regarding orchestration which consists of 13.4% of DevOps topics we found. Furthermore, their analysis missed Infrastructure as Code, a very important concept of DevOps having 28.9% of discussions in Stack Overflow.

**Study on Software Release Engineering.** Another recent paper [12] performs a similar empirical study on release engineering questions in Stack Over-

flow, using LDA-based topic modeling techniques. As explained in the background section, Release engineering can be seen an important but smaller subset of DevOps. This is also evident since the topics found in the release engineering paper [12] are clearly separated from most of the DevOps topics found in our empirical study.

The release engineering paper reports 'Mobile Deployment' as a topic, whereas our paper discussed 'Build Mobile App' as a DevOps topic. It seems that their focus was more on the delivery side concerns such as deployment or build failure, whereas we found unique DevOps topics covering full build lifecycle from Git Build, Build Project or end to end Jenkins Pipeline. Moreover, Jenkins was not considered as any standalone topic there and continuous integration comprised of only 3.3% discussion, whereas we reported a massive 21.4% SO posts only regarding Jenkins which is the most prominent continuous integration tool.

In spite of focusing on release engineering, their paper did not report any topic regarding Kubernetes, the state-of-the-art orchestration tool, whereas we reported that Kubernetes holds a major part, around 13.4%, of all DevOps questions. In release engineering, the authors mostly presented traditional testing and infrastructure or configuration management without automation, whereas in DevOps concept, we found that 18% of posts are more concerned in automation aspects such as configuration automation (8.6%), cloud infra automation (6.2%), functional test automation (3.2%).

Since management perspective is one core concept of DevOps, our paper discussed organizational issue such as agile development process and developers' training for DevOps skill development, whereas such perspective is missing in the release engineering study.

All the above studies are limited to the building, integration, and deployment process and to the best of our knowledge there is yet to be a study on the overall process, people, delivery and runtime aspects of DevOps. The summary of the differences along with our research contribution is provided in the Table 10 and Table 11. In this paper, we report a mixed methods study on holistic view of DevOps concepts and on challenges faced by engineers by analyzing 174K SO

Table 10: Comparison with different DevOps related study types

<b>Study Type</b>	<b>Previous Study</b>	<b>DevOps Topic</b>	<b>Research Contribution</b>
Definition and Literature Survey	“DevOps dimension by Lwakatare et al. [62] DevOps definition by Aeillo et al. [15] Literature survey by Leite et al. [13] Mapping study by Jabbari et al. [60]”	DevOps Definition	Our research is based on the definition provided by these research outcomes. Specifically, we performed open coding of our survey responses on the basis of DevOps aspects (management, engineering, people and process) defined by Leite et al.
Case studies	Case study by Luz et al. [61], by Smeds et al. [63], by Senapathi et al. [64], by Lwakatare et al. [4], by Erich et al. [66], by Guerriero et al. [65].	Adoption and Practice	These are specific studies performed on the case specific struggles faced by the companies. However, in our research, we also cover the challenges in the industry through a qualitative survey. Our survey has provided insight on industry wide challenges on skill shortage, and lack of organizational training.
	Research by Schermann et al. [67], by Haque et al. [68]	Docker Discussion	Specific studies performed on Docker tool only. Our research finds that Docker discussion comprises of only 2.7% of all DevOps discussions.
SE Research	Study by Rahman et al. [71] focuses mainly on Puppet, a major tool of Infrastructure as Code (IaC)	IaC Discussion	In our research, we found that there are other IaC tools such as TerraForm, CloudFormation which are gaining popularity in discussion (having 6.2% of questions in SO). Overall, Puppet and Chef combined is discussed 4.6% times among all DevOps topics in Stack Overflow.

posts and by conducting a survey of 21 DevOps professionals in the industry.

## 8. Conclusion

In this mixed methods empirical research, we have applied topic modeling on 174K Stack Overflow posts that contain DevOps related discussions. We have found 23 DevOps topics grouped into four categories. The topic category ‘Cloud & CI/CD Tools’ contains the most number of topics (10) which cover 48.6% of all questions in our dataset, followed by the category Infrastructure as Code (28.9%). The file management and script execution topics from the category Infrastructure as code are the most popular (with most views), while

Table 11: Additional Contribution to CSE and RE domain related researches

Domain	Previous Study	DevOps Topic	Research Contribution
Continuous Software Engineering	Research by Zahedi et al. [39] focused on the engineering process aspects of DevOps only. They explicitly tried to avoid the DevOps related tools from the dataset by excluding related tags in Stack Overflow.	CI/CD Discussion	<p>Our empirical dataset is 13 times larger than this study. Moreover, this study has limitation for excluding the tools related tags in SO. Also, unlike ours, this study does not provide grouped categories of the 32 CI/CD related topics. As a result, this is very specific and granular level research focusing only on the CI/CD process. However, in our study, we have grouped 23 DevOps topics under 4 major categories. Also, we found that CI/CD tools constitutes of 33.2% of all DevOps discussions. So, topic coverage wise this study focuses on less than one third of our research area and the topics are very granular.</p> <p>The study did not cover infrastructure as code (cloud and configuration automation), orchestration (Kubernetes), and automated testing concepts which consists of almost 48.6% DevOps related discussion.</p>
Release Engineering	Relevant study by Openja et al. [12] focused on the release engineering steps related discussions in Stack Overflow till December 2017.	Release Engineering Discussion	<p><b>Firstly</b>, the focus area of this study only covers a portion of total DevOps concept. For example, because they only mined ReLeng topics, Continuous Integration came in 3.3% of their posts whereas we found that Continuous Integration tool Jenkins was discussed in 21.4% of all DevOps posts. They found highest posts in Software Testing (5.2%) since they only considered the Release phase of the SDLC, but in total DevOps we found that Quality Assurance (functional and non-functional) posts are more frequent (6.3%). <b>Secondly</b>, since their study is based on almost 3 years old data (December 2017 SO dump vs ours July 2020 SO dump), many of the latest trends have been missed; for example, we found that Infrastructure as Code is discussed in 28.9% of posts, whereas according to their findings it was only 11.1% of ReLeng discussion. <b>Finally</b>, we extended our research on the people, organizational, and process aspects of DevOps [13] by conducting a qualitative explanatory survey and identified skill, and training related hurdles in DevOps whereas their study only performed empirical data analysis.</p>

infrastructural Exception Handling, and non-functional test automation have the highest percentage of unsolved questions. We validate the quantitative study by and provide qualitative analysis on DevOps challenges by conducting a survey among 21 industry professionals. Our findings can be used by the vendors of DevOps tools to address the issues reported by practitioners; by architects and managers for system design and DevOps adoption decisions. In future, large scale exploratory survey can be conducted to gain insights for the solution on DevOps challenges.

## References

- [1] L. Bass, I. Weber, L. Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley Professional, 2015.
- [2] M. Loukides, What is DevOps?, O'Reilly Media, Inc, 2012.
- [3] Upguard, Devops stats for doubters (2021).  
URL <https://www.upguard.com/blog/devops-success-stats>
- [4] L. E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä, J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo, C. Lassenius, Devops in practice: A multiple case study of five companies, *Information and Software Technology* 114 (2019) 217 – 230.
- [5] J. Roche, Adopting devops practices in quality assurance: Merging the art and science of software development, *ACM Queue* 111 (9) (2013) 20 – 27.
- [6] P. Labs, 2015 State of the DevOps Report, <https://puppet.com/resources/report/2015-state-devops-report>, last accessed on 15 January 2022 (2015).
- [7] S. Overflow, Stack Overflow Questions, <https://stackoverflow.com/questions/>, last accessed on 14 November 2020 (2020).
- [8] M. Bagherzadeh, R. Khatchadourian, Going big: A large-scale study on what big data developers ask, in: Proceedings of the 2019 27th ACM Joint

Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2019, ACM, New York, NY, USA, 2019, pp. 432–442.

- [9] S. Ahmed, M. Bagherzadeh, What do concurrency developers ask about? a large-scale study using stack overflow, in: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM ’18, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3239235.3239524.  
URL <https://doi.org/10.1145/3239235.3239524>
- [10] Z. Wan, X. Xia, A. E. Hassan, What is discussed about blockchain? a case study on the use of balanced lda and the reference architecture of a domain to capture online discussions about blockchain platforms across the stack exchange communities, IEEE Transactions on Software Engineering (2019).
- [11] A. Bandeira, C. A. Medeiros, M. Paixao, P. H. Maia, We need to talk about microservices: an analysis from the discussions on stackoverflow, in: 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), IEEE, 2019, pp. 255–259.
- [12] M. Openja, B. Adams, F. Khomh, Analysis of modern release engineering topics:—a large-scale study using stackoverflow—, in: 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, 2020, pp. 104–114.
- [13] L. Leite, C. Rocha, F. Kon, D. Milojicic, P. Meirelles, A survey of devops concepts and challenges, ACM Computing Surveys 52 (6) (2020) 1 – 35.
- [14] A. of the Study, Replication package of the study (2023).  
URL <https://github.com/DevOpsTopic/Challenges>
- [15] B. Aiello, L. Sachs, Agile Application Lifecycle Management: Using DevOps to Drive Process Improvement, Addison-Wesley Professional, 2016.

- [16] B. Fitzgerald, K.-J. Stol, Continuous software engineering and beyond: trends and challenges, in: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering, 2014, pp. 1–9.
- [17] A. Dyck, R. Penners, H. Licher, Towards definitions for release engineering and devops, in: 2015 IEEE/ACM 3rd International Workshop on Release Engineering, 2015, pp. 3–3. doi:[10.1109/RELENG.2015.10](https://doi.org/10.1109/RELENG.2015.10).
- [18] B. Adams, S. McIntosh, Modern release engineering in a nutshell – why researchers should care, in: 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), Vol. 5, 2016, pp. 78–90. doi:[10.1109/SANER.2016.108](https://doi.org/10.1109/SANER.2016.108).
- [19] M. Van Antwerp, G. Madey, The importance of social network structure in the open source software developer community, in: 2010 43rd Hawaii International Conference on System Sciences, IEEE, 2010, pp. 1–10.
- [20] Stack Exchange 2020-06. Stack Exchange Data Dump download, <https://www.brentozar.com/go/querystack> (2020).
- [21] X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, J.-L. Sun, What security questions do developers ask? a large-scale study of stack overflow posts, Journal of Computer Science and Technology 31 (5) (2016) 910–924.
- [22] S. Ahmed, M. Bagherzadeh, What do concurrency developers ask about?: A large-scale study using stack overflow, in: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2018, p. Article No. 30.
- [23] M. Bagherzadeh, R. Khatchadourian, Going big: a large-scale study on what big data developers ask, in: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2019, pp. 432–442.
- [24] T. Mens, A. Serebrenik, A. Cleve, Evolving Software Systems, Vol. 190, Springer, 2014.

- [25] T.-H. P. Chen, S. W. Thomas, A. E. Hassan, A survey on the use of topic models when mining software repositories, *Empirical Software Engineering* 21 (5) (2016) 1843–1919.
- [26] X. Sun, X. Liu, B. Li, Y. Duan, H. Yang, J. Hu, Exploring topic models in software engineering data analysis: A survey, in: 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2016, pp. 357–362.
- [27] B. A., T. S.W., A. Hassan, What are developers talking about? an analysis of topics and trends in stack overflow,, in: analysis of topics and trends in stack overflow, *Empir. Softw. Eng.* 19 (3) (2012) 619–654., 2012.
- [28] X. Bowen, X. Zhenchang, X. Xin, L. David, What security questions do developers ask? a large-scale study of stack overflow posts., In Proceedings of the 32Nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2017). IEEE Press, Piscataway (2019).
- [29] NLTK, Sentiment Analysis, <http://www.nltk.org/howto/sentiment.html> (2016).
- [30] R. Řehůřek, P. Sojka, Software framework for topic modelling with large corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010, pp. 45–50.
- [31] C.-M. Tan, Y.-F. Wang, C.-D. Lee, The use of bigrams to enhance text categorization, *Information processing & management* 38 (4) (2002) 529–546.
- [32] M. F. Porter, An algorithm for suffix stripping, 1st Edition, Morgan Kaufmann Publishers Inc., 1997.
- [33] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research* 3 (4-5) (2003) 993–1022.

- [34] A. K. McCallum, Mallet: A machine learning for language toolkit, <http://mallet.cs.umass.edu/>, as of Feb 24, 2012 (2002).
- [35] L. R. Biggers, C. Bocovich, R. Capshaw, B. P. Eddy, L. H. Etzkorn, N. A. Kraft, Configuring latent dirichlet allocation based feature location, *Journal Empirical Software Engineering* 19 (3) (2014) 465–500.
- [36] W. Hudson, The Encyclopedia of Human-Computer Interaction, 2nd Edition, The Interaction Design Foundation, 2013, Ch. Card Sorting.
- [37] S. Fincher, J. Tenenberg, Making sense of card sorting data, *Expert Systems* 22 (3) (2005) 89–93.
- [38] A. Barua, S. W. Thomas, A. E. Hassan, What are developers talking about? an analysis of topics and trends in stack overflow, *Empirical Software Engineering* (2012) 1–31.
- [39] M. Zahedi, R. Rajapakse Namal, M. Babar Ali, Mining questions asked about continuous software engineering: A case study of stack overflow., in: In Proceedings of the Evaluation and Assessment in Software Engineering (Trondheim, Norway) (EASE '20)., 2020.
- [40] C. Rosen, E. Shihab, What are mobile developers asking about? a large scale study using stack overflow, *Empirical Software Engineering* (2015) 33.
- [41] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, E. Shihab, Challenges in chatbot development: A study of stack overflow posts, in: 17th International Conference on Mining Software Repositories, October 5–6, 2020, Seoul, Republic of Korea. New York, NY, USA, ACM, 2020.
- [42] M. G. Kendall, A new measure of rank correlation, *Biometrika* 30 (1/2) (1938) 81–93.
- [43] W. H. Kruskal, Historical notes on the wilcoxon unpaired two-sample test, *Journal of the American Statistical Association* 52 (1957) 356–360.

- [44] W. P. Vogt, R. B. Johnson, *The SAGE dictionary of statistics & methodology: A nontechnical guide for the social sciences*, Sage publications, 2015.
- [45] J. W. Creswell, J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*, Sage publications, 2017.
- [46] M. Miles, A. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*, SAGE Publications, 1994.
- [47] J. Cohen, A coefficient of agreement for nominal scales, *Educational and Psychological Measurement* 20 (1) (1960) 37–46.
- [48] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, K. A. Schneider, Answering questions about unanswered questions of stack overflow, in: Proceedings of the 10th Working Conference on Mining Software Repositories, 2013, pp. 87–100.
- [49] B. Xu, Z. Xing, X. Xia, D. Lo, Answerbot: automated generation of answer summary to developers' technical questions, in: Proc. 32nd IEEE/ACM International Conference on Automated Software Engineering, 2017, pp. 706–716.
- [50] T. Zhang, G. Upadhyaya, A. Reinhardt, H. Rajan, M. Kim, Are code examples on an online q&a forum reliable?: a study of api misuse on stack overflow, in: In Proceedings of the 40th International Conference on Software Engineering, 2018, pp. 886–896.
- [51] D. Yang, A. Hussain, C. V. Lopes, From query to usable code: an analysis of stack overflow code snippets, in: In Proceedings of the 13th International Conference on Mining Software Repositories, 2016, pp. 391–402.
- [52] V. Terragni, Y. Liu, S.-C. Cheung, Csnippex: automated synthesis of compilable code snippets from q&a sites, in: In Proceedings of the 25th International Symposium on Software Testing and Analysis, 2016, pp. 118–129.

- [53] X. Ren, Z. Xing, X. Xia, G. Li, J. Sun, Discovering, explaining and summarizing controversial discussions in community q&a sites, in: 34th IEEE/ACM International Conference on Automated Software Engineering, 2019, pp. 151–162.
- [54] L. Ponzanelli, A. Mocci, A. Bacchelli, M. Lanza, Improving low quality stack overflow post detection, in: In Proceedings of the 30th International Conference on Software Maintenance and Evolution, 2014, pp. 541–544.
- [55] M. Bagherzadeh, R. Khatchadourian, Going big: A large-scale study on what big data developers ask, in: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2019, ACM, New York, NY, USA, 2019, pp. 432–442.
- [56] C. Rosen, E. Shihab, What are mobile developers asking about? a large scale study using stack overflow, *Journal Empirical Software Engineering* 21 (3) (2016) 1192–1223.
- [57] X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, J.-L. Sun, What security questions do developers ask? a large-scale study of stack overflow posts, *Journal of Computer Science and Technology* 31 (5) (2016) 910–924.
- [58] P. Labs, 2021 State of the DevOps Report, <https://media.webteam.puppet.com/uploads/2021/07/Puppet-State-of-DevOps-Report-2021.pdf>, last accessed on 15 January 2022 (2021).
- [59] P. Labs, A Decade of DevOps, <https://puppet.com/devops/state-of-devops-report-retrospective/>, last accessed on 15 January 2022 (2021).
- [60] R. Jabbari, N. bin Ali, K. Petersen, B. Tanveer, What is devops? a systematic mapping study on definitions and practices, XP '16 Workshops, Association for Computing Machinery, New York, NY, USA, 2016.

doi:10.1145/2962695.2962707.

URL <https://doi.org/10.1145/2962695.2962707>

- [61] W. P. Luz, G. Pinto, R. Bonifácio, Adopting devops in the real world: A theory, a model, and a case study, *Journal of Systems and Software* 157 (2019) 110384.
- [62] L. E. Lwakatare, P. Kuvaja, M. Oivo, Dimensions of devops, in: International conference on agile software development, Springer, 2015, pp. 212–217.
- [63] J. Smeds, K. Nybom, I. Porres, Devops: a definition and perceived adoption impediments, in: International conference on agile software development, Springer, 2015, pp. 166–177.
- [64] M. Senapathi, J. Buchan, H. Osman, Devops capabilities, practices, and challenges: Insights from a case study, *EASE’18*, Association for Computing Machinery, New York, NY, USA, 2018, p. 57–67. doi:10.1145/3210459.3210465.  
URL <https://doi.org/10.1145/3210459.3210465>
- [65] M. Guerriero, M. Garriga, D. A. Tamburri, F. Palomba, Adoption, support, and challenges of infrastructure-as-code: Insights from industry, in: 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2019, pp. 580–589. doi:10.1109/ICSME.2019.00092.
- [66] F. M. Erich, C. Amrit, M. Daneva, A qualitative study of devops usage in practice, *Journal of software: Evolution and Process* 29 (6) (2017) e1885.
- [67] J. Cito, G. Schermann, J. E. Wittern, P. Leitner, S. Zumberi, H. C. Gall, An empirical analysis of the docker container ecosystem on github, in: In 2017 IEEE/ACM14th International Conference on Mining Software Repositories (MSR), 2020.
- [68] M. UL Haque, L. Horn Iwaya, M. Babar Ali, Challenges in docker development: A large-scale study using stack overflow, in: ESEM ’20: Proceedings

of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2020.

- [69] J. Bosch, Continuous software engineering: An introduction, in: Continuous software engineering, Springer, 2014, pp. 3–13.
- [70] B. Fitzgerald, K.-J. Stol, Continuous software engineering: A roadmap and agenda, *Journal of Systems and Software* 123 (2017) 176–189.
- [71] A. Rahman, A. Partho, P. Morrison, L. Williams, What questions do programmers ask about configuration as code?, in: Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering, 2018, pp. 16–22.