

Understanding DevOps Critical Success Factors: A Thematic Analysis

Azad Nasren, Hyrynsalmi Sami, Smolander Kari

This is a Author's accepted manuscript (AAM) version of a publication
published by Springer, Cham

in Digital Product Management in the Era of Data Economy, Artificial Intelligence, and
Ecosystems. ICDPM 2024. Lecture Notes in Business Information Processing

DOI: 10.1007/978-3-031-71515-0_3

Copyright of the original publication:

© 2025 The Author(s), under exclusive license to Springer Nature Switzerland AG

Please cite the publication as follows:

Azad, N., Hyrynsalmi, S., Smolander, K. (2025). Understanding DevOps Critical Success Factors: A Thematic Analysis. In: Petrik, D., Saltan, A., Helferich, A. (eds) Digital Product Management in the Era of Data Economy, Artificial Intelligence, and Ecosystems. ICDPM 2024. Lecture Notes in Business Information Processing, vol 528. Springer, Cham. https://doi.org/10.1007/978-3-031-71515-0_3

**This is a parallel published version of an original publication.
This version can differ from the original published article.**

Understanding DevOps Critical Success Factors: A thematic analysis

Nasreen Azad^[0000-0002-2428-2984], Sami Hyrynsalmi^[0000-0002-5073-3750], and
Kari Smolander^[0000-0002-7043-0458]

LUT University, Finland
{nasren.azad, sami.hyrynsalmi, kari.smolander}@lut.fi

Abstract. The software development industry has widely adopted DevOps culture and practices, which emphasize the importance of effective utilization of DevOps tools, promoting collaboration, developing necessary skills among employees, and fostering a supportive company culture. Understanding the critical success factors for DevOps is essential for ensuring the overall performance of an organization. To gain a deeper understanding of DevOps practices and insights from professionals during the adoption of DevOps within teams, a study was conducted aimed at identifying DevOps critical success factors for teams within software-producing organizations. The study employed a thematic approach through collaborative coding and extensive interviews with 16 DevOps professionals in the IT industry. The findings highlighted the importance of fostering a (1) collaborative organizational culture, (2) Integrating DevOps automation, (3) cultivating strong work ethics, (4) addressing challenges related to remote teamwork, and (5) establishing dedicated DevOps security teams. Our findings thus contribute to improving DevOps concepts of success factors, practices, and organizational impacts.

Keywords: DevOps, Critical Success Factors, Software Development, Software Operations, Thematic analysis, Qualitative research

1 Introduction

As technology evolves, the software development industry must rapidly adapt to emerging trends to remain competitive and relevant in the market [39]. Prioritizing product quality, efficient testing, timely releases, and swift time-to-market are crucial for success in this dynamic landscape [25].

The adoption of DevOps has surged in recent years, with substantial investments from IT organizations and software industries [19]. The global DevOps market is projected to experience robust growth, indicating its increasing importance [19]. However, successful DevOps adoption impacts significant cultural and organizational shifts within IT departments [35, 8]. The research conducted by Shahin et al. [41] identified seven critical success factors (CSFs) essential for continuous practices, including testing, transparency, team awareness, design principles, customer environment, a motivated and highly skilled team, accurate

infrastructure, and application domain. However, these factors lack explicit validation. Dikert et al. [20] highlighted the need for additional research to identify challenges and determine the most significant success factors for agile transformations, emphasizing the importance of discerning genuine critical relevance for organizations [8].

Scholars have explored the impact of CSFs on organizational performance, emphasizing the intricate relationship between successful implementation and organizational improvement [40, 6, 4, 8]. Linking organizational success to identified CSFs is crucial for effective measurement, encompassing tangible benefits like cost reduction and intangible benefits such as customer satisfaction and project success [9, 8].

The implementation and adoption of DevOps has witnessed significant growth in recent years in terms of investment for IT organizations and software industries, with most surveys indicating a notable surge in DevOps adoption [19]. According to the DevOps Trends Survey study 2023, the global DevOps market is projected to experience a robust compound annual growth rate of approximately 20 to 25 percent until 2030 [19]. Despite this upward trend, achieving successful DevOps adoption remains contingent on profound cultural and organizational shifts within companies' IT departments. However, organizations often face challenges when formulating a DevOps adoption strategy and establishing effective team structures. This difficulty in defining a coherent strategy and structuring teams successfully can adversely affect team performance and success [15, 8].

Despite increasing interest and academic inquiry into DevOps, a comprehensive understanding of critical success factors for its effective implementation remains elusive [23, 4, 7, 5]. In response to this research gap, we have applied thematic analysis based on 16 professional interviews to understand various success factors and their relationship with DevOps practices. To address the aim of this research we have one research question (RQ).

RQ What is the significance of Critical Success Factors (CSF) in DevOps practices within IT organizations?

This research paper makes three key contributions. Firstly, it enhances understanding of DevOps success factors by synthesizing existing literature and interview data. Secondly, it offers detailed insights into DevOps implementation and its impact on organizational performance. Lastly, it provides practical recommendations for organizations to improve their DevOps practices, bridging theory with actionable guidance for practitioners.

The remaining of the study is structured as follows. Section 2 presents the empirical research strategy, followed by results in Section 3. Section 4 summarizes the essential findings, Section 5 presents the discussion, Section 6 explains related work, and Section 7 presents the study's conclusion.

2 Research Process

2.1 Data collection

We conducted semi-structured interviews, c.f. [26], with 16 DevOps professionals from the software development industry actively engaged in DevOps practices. Each interview session lasted approximately 45 minutes, during which we obtained permission from the participants to record the discussions. In total, we dedicated 12 hours to conducting these interviews, during which the respondents generously shared valuable insights drawn from their current and past experiences in DevOps. The extensive discussions yielded substantial data, resulting in 480 pages of transcribed documents for subsequent analysis. The interviews were conducted in the second quarter of 2022 and involved professionals from five countries: Finland, Sweden, Norway, Germany, and the United Kingdom. We employed various recruitment methods to ensure a diverse pool of participants, specifically those working in DevOps. Initially, we leveraged personal contacts and professional networks on social media platforms such as Twitter and LinkedIn to identify potential candidates. Subsequently, we reached out to them via email and private messages to gauge their interest in participating in our study and sharing their insights.

Furthermore, we utilized the snowball recruitment technique to expand our participant pool [47, 46]. This method allowed us to identify additional professionals through referrals from initial interviewees, thereby enriching the diversity of perspectives in our research. The interviews were scheduled based on the availability of the participants. They were conducted in teams using a predefined set of guiding questions that covered our research’s main areas of interest [45].

2.2 Data extraction and analysis

Data was first coded based on interviews, and later on, the data synthesis was done [29]. The analysis process was initiated using a thematic coding technique outlined by Braun and Clarke [16, 17]. The researchers extracted success factors and professionals’ perspectives regarding DevOps practices from the interview quotes and applied them to the analysis. Firstly, the researchers aimed to identify critical success factors for DevOps practices. Secondly, they tried to understand professionals’ perspectives on DevOps practices according to success factors. Thirdly, they tried to find connections between DevOps’ success factors and organizational practices. The researchers developed a set of predefined themes from the transcripts. They analyzed and discussed the results and followed a strict process to resolve disagreements. The researchers familiarized themselves with the data collection, engaged in in-depth discussions based on the interviews, and went through a rigorous process to analyze the data.

We started gathering data from the professional’s mentions of success factors and organizational practices in the transcripts. For Example, *“Initially, we relied on GitHub and Jenkins as our primary tools. However, DevOps represents a cultural shift in product and service companies’ operations. In the DevOps*

| ID | Role in Com | Domain | Team location | Team types | Team members | Country | No. of employees in the comp. | em-Tools in | Sw.dec. method | Exp in SWD (yr) |
|-----|---------------------------|------------------------------------|------------------|-----------------------------------|--------------|---------|-------------------------------|-------------------------------------|----------------|-----------------|
| R1 | Cloud Engineer | IT | Co-located | Developers | 14 | Finland | 571(s) | Docker Git-10 | | |
| R2 | Cloud Engineer | Energy com-pany | Co-located | Developers | 10 | Finland | 19000 (L) | Lab Scrum GitLab Ku-N/A | | 3 |
| R3 | Site Reliability Engineer | Semiconductor and Software | Co-located | Developers | 20 | UK | 6210(M) | Docker Azure CI/CD | Kanban | 8 |
| R4 | DevOps consultant | company DevOps consulting com-pany | Distributed | Developers | 8 | Germany | not mentioned | men-GitLab, Docker, Anst- CI/CD, | Scrum, | 10 |
| R5 | DevOps Engineer | Digital Branding company | Co-located | Full stack developer | 10 | Norway | 6000(M) | GitLab, Azure, Ter-Scrum | Kanban, | 4.5 |
| R6 | Software Developer | Financial Consultant | Co-located | Front-end and back-end Developers | 10 | Finland | JIRA CI /CD GitHub | Circle CI | 7 | |
| R7 | Technical PM | Global cyber security | Co-located | Developers | 15 | Finland | 1656(s) | GitHub Docker Jira | Scrum | 9 |
| R8 | CTO | Digital lets | Wal- Distributed | DevOps Engi-ners | 7 | Sweden | 15(S) | Kanban GitHub, Azure, Aws ban | scrum, Kan- | 16 |
| R9 | Tech lead | DevOps com-pany | Co-locate | DevOps engi-ners | 8 | Finland | 38(S) | JIRA, Ter-Kanban | raform, | 10 |
| R10 | DevOps Engineer | Business | co-located | Developer | 10 | Germany | 4238(M) | Azure Docker Ter-Scrum | | 7 |
| R11 | Cloud Engineer | DevOps com-pany | Co-located | Full stack developer | 7 | Finland | 1389(S) | raform JIRA Docker Ban- | Kanban | 8 |
| R12 | CTO | IT Security | co-located | DevOps Engi-ner | 13 | Finland | 590(S) | boo Docker GitHub AWS Terraform | Kanban | 5 |
| R13 | Jr. software developer | Insurance company | Distributed | DevOps Engi-ner | 8 | Sweden | 1400(S) | GitLab Kibana Jenk- ins Terraform | scrum | 6 |
| R14 | DevOps Engineer | IT company | Co-located | DevSecOps engineer | 10 | Sweden | 2000(M) | Ansible Ban-boo Docker Jenkins Lira | scrum | 9 |
| R15 | Senior software developer | DevOps com-pany | Distributed | DevOps Engi-ner | 8 | Norway | 278(S) | Gitlab | Kanban | 6 |
| R16 | DevOps Engineer | IT Consult- ing company | Distributed | DevOps Engi-ner | 6 | UK | 943(S) | GitHub Docker Ku-bernetes Ansible | Kanban | 10 |

Table 1: Participant’s Demographic information and working practices, SN represents Respondents from R1–R16, Respondents’ role in the company, Domain, Team location, Team types, Team members in the team, Countries, and Number of employees

approach, developers are more actively engaged with customers and have greater exposure to real customer problems. For me, the essence of DevOps lies in automation and the effective use of tools”- (R8) is an example of how good tooling helps developers to perform better for customers’ needs through automation. Similarly, the quote, *“Currently, in most applications, this is infrastructure as code because nowadays, everything goes into the cloud. We are no longer using one-premise data centers. So, our main goal is to clarify everything. If we want to deploy everything into the cloud, it gets messy. It’s also time-consuming, and we want to automate things. so, we use infrastructure as a code which speeds up the automation process.”-(R2)* denotes a professional’s recommendation where using infrastructure as code for DevOps makes the automation process work faster and more efficiently.

We tried to group two similar mentions equally regarding the main idea shared in the interviews. So, we analyzed those two similar mentions as one idea. Here are example quotes from the analysis:

“Well, there is always room for improvement. I think the main thing is to understand development operations and security. We want to include or if you want to talk about like DevSecOps or DevOps. So, we need to understand each other’s perspective”-(R2). *“There are always challenges in the software development process. If we integrate security from the beginning, it would be better for the process. So, I saw that people don’t understand the importance of security. The whole thing should be conducted because it’s not only just developing, it’s also like securing the applications for the long run”-(R8).*

In the above example, both quotes refer to the importance of implementing security from the start of the development process. We use the thematic analysis framework proposed by Braun and Clarke [16] to create themes that summarize the data. During the familiarization phase, the researchers take the necessary steps to become familiar with the data. To do this, researchers review the transcriptions thoroughly.

We meticulously analyzed the data we collected. We assigned unique codes to the data entries, briefly describing the interviewees’ words. These codes were used to categorize and organize the data for analysis. To ensure a fair and accurate analysis, the researchers divided the work into three parts, allowing each author to focus on specific aspects of the data. Each author followed a detailed procedure outlined at the beginning of the subsection to ensure consistency in the analysis process. This methodical approach allowed the researchers to analyze the data and draw meaningful conclusions.

The researchers start to sort the codes into themes. Some themes might be sub-themes of others, or some codes can become themes if they are pertinent. In Figure 1, we show how codes, themes, and categories are connected and emerge from the data. In Figure 1, we give an example of the coding process. In the figure, there are 3 example themes *“Development culture and mindset”*, *“DevOps working Ethics”*, and *“Tools for supporting DevOps”*. In the coding phase of *“Development culture and mindset”*, there are four codes named *“Having a collaboration in teams”*, *“Distributing the responsibilities”*, *“Knowledge sharing in teams”*, and *“DevOps*

perception". In the "Ethical issues for DevOps" theme has five codes naming "Leadership issues in teams", "Toxic culture in workplace", "Discrimination in teams", "Unrealistic goals" and "Unethical use of company tools". In the third theme of "Tools for supporting DevOps", there are four codes naming "Continuous integration", "Automation", "Continuous monitoring", "Continuous testing and delivery". All three categories are connected to the concept of implementation of DevOps practices.

A discrepancy between different success factors, professional recommendations, and organizational performance can be attributed to an inconsistency. This inconsistency can occur because the challenges and recommendations reflect the specific DevOps practices being used, which may be different from the experiences of others.

3 Results and findings

3.1 Themes

Collaboration and DevOps culture Every organization has a team culture that differs from one organization to another. DevOps culture aims to create collaboration between the developers and Ops people [8, 4, 7, 6, 9, 2]. During our interviews; we got much insightful information about DevOps culture in organizations and how in practice it works for both teams. This collaboration culture impacts CSF and company performance. One respondent quoted that,

"There are several very experienced team members. Due to that, sometimes there is not much knowledge gap there, but we have resource gap like we need more people" - R1 (Cloud Engineer, Finland).

A professional from an IT company stated that *"Being like in culture, where everyone contributes and everyone kind of understands each other's point of view. We want to break this wall or silos so that we make the software development process faster"* - R2 (Cloud Engineer, Finland)

DevOps and working Ethics Ethical issues in software development are very crucial in terms of organizational, social, and technical contexts [5]. Ethics has a role in organizations' performance which also connects critical success factors to some extent [5]. Software developers should consider ethics while working on the software development lifecycle [43, 10]. When there is an unethical software issue it is followed by many consequences depending on who the users or audience of the software usage [43].

According to a professional *"To handle work ethics when working in a team, it's best to have a small, integrated team where everyone is engaged in the development process from start to finish. This enables developers and teams to facilitate ethical conversations and eliminate toxic culture and poor leadership."* - R15 (Senior Software Developer, Norway).

Table 2: The propositions identified in this study.

| ID | Identified proposition |
|-----|--|
| P1 | Team collaboration facilitates the Development culture and mindset of teams |
| P2 | Responsibility distribution facilitates the Development culture and mindset of teams |
| P3 | Knowledge sharing facilitates the Development culture and mindset of teams |
| P4 | DevOps perception might facilitate the Development culture and mindset of teams |
| P5 | Leadership issues in teams decrease teams' ethical working environment |
| P6 | Toxic culture in the workplace decreases ethical working environment in teams |
| P7 | Discrimination in teams decrease ethical working environment in teams |
| P8 | Unethical use of tools decreases ethical working environment in teams |
| P9 | Continuous integration increases DevOps implementation practices |
| P10 | Automation increases DevOps implementation practices |
| P11 | Continuous monitoring increases DevOps implementation practices |
| P12 | DevOps implementation practices support quality assurance for the product |
| P13 | DevOps implementation practices support continuous measurement for DevOps teams |
| P14 | DevOps implementation practices support collaborative culture for the teams |
| P15 | DevOps implementation practices are enhanced with DevOps security |
| P16 | Remote work negatively decreases DevOps implementation practices |

DevOps automation DevOps automation is a practice or discipline of using different software tools and methodologies to automate some repetitive and manual tasks to make the software process faster [11, 3]. By adopting automation, a company can reduce risks and streamline the process. Some teams can adopt the automation process and get an advantage while testing codes to make the deployment [9, 4]. Some companies partially use automation for specific tasks. For better software performance teams should automate most of the processes [13].

According to a developer “*Because it is possible that you just push the code, someone will review the code and after that, you don’t have to do anything. If there is 100 percent automation, You just wait and it should be in production after the auto-continuous testing and continuous integration. All of these tasks will be done automatically, So the developer has fewer things to handle*” - R1 (Cloud Engineer, Finland)

Remote teamwork challenge. Software engineering is a highly technical, knowledge-oriented task that combines collaboration and coordination with developers and stakeholders. This task also requires a very focused and uninterrupted work process from developers’ commitment so that they can efficiently develop good quality and efficient software for the software industry [37]. Very often software is developed by open-source communities, remotely, and distributed software projects are globally distributed [22].

A tech lead mentioned that “*Remote work requires tools for developers to produce code, build and deploy software from anywhere. Embedded tools help teams communicate and perform efficiently, eliminating poor communication and time zone challenges*” - R9 (Tech Lead, Finland).

DevOps security teams DevSecOps is a security testing integration practice that is involved in every stage of the software development process [38]. This encourages collaboration between developers, security specialists, and operations teams so that the software is built efficiently and secured for the system with the help of

tools and processes [38]. DevSecOps helps in the cultural transformation which makes security a shared responsibility for those who are involved in building the software. In DevSecOps, security takes place earlier in the development cycle, and it is already in the planning phase before the deployment is done for production [49].

IT professionals stated that *"The importance of DevSecOps in the development process is crucial. There are some advantages to having a security check earlier. It reduces human errors by automating security testing. Thus, the security assessment prevents the bottleneck situation in the development process"*- R7 (Technical Project Manager, Finland).

3.2 Towards a Theory for DevOps CSF

This section summarizes the CSF theory with the recommendations and reporting from software engineering disciplines [42]. While describing a theory, it should have constructs, propositions, scope, and explanations [36]. Here, we propose a sample of CSF theory that influences different organizational issues. DevOps critical success factors are the central concept that helps build a company culture between development and operation teams. DevOps CSF is supported by other concepts, including Development culture and mindset. In Figure 1, the first box on the left represents Development culture and mindset. For example, Collaboration in teams, responsibility distribution, Knowledge sharing, and DevOps perception represent the concept of development culture and mindset. The second box on the left represents Ethical issues for DevOps meaning Leadership issues, Toxic culture in the workplace, discrimination in teams, unethical use of tools, etc). The last box on the left represents the concept of tools for supporting DevOps, which means continuous integration, automation, and monitoring. On the right side in Figure 1 assuring quality, continuous measure, collaboration culture, DevOps security teams, and remote teamwork impacts on DevOps organizational performances. Based on categories and concepts, relationships of subcategories, and transcription memos we have identified 16 propositions. In Figure 2 the proposed propositions and relationships are shown.

In Figure 2 the theory elements are represented and the relationships between development teams and operation teams are highlighted. According to our findings, DevOps implementation practices support quality assurance (P12), Continuous measure (P13), and collaboration culture (P14) in IT organizations. DevOps security (P15) is good for DevOps implementation practices and is enhanced with DevOps security. A good security team's involvement from the beginning of the process gives more secure applications for the users. For the proposition of remote work (P16), we have found that Remote work decreases DevOps implementation practices, and the engineers suggested that remote work decreases confidence and communication problems and improves the development process. Team collaboration (P1), responsibility distribution (P2), knowledge sharing (P3), and DevOps perception(P4) might facilitate the Development culture and mindset of teams. We have also found that leadership issues (P5), toxic culture (P6), discrimination in teams (P7), and unethical use of tools (P8) in teams decrease

teams' ethical working environment, and that makes the project fail. Continuous integration (P9), automation (P10) and continuous monitoring (P11) increase DevOps implementation practices in teams.

4 Discussion

4.1 Key findings

Insights into Critical Success Factors can significantly impact DevOps practices in IT organizations by providing a framework for understanding and prioritizing key elements that contribute to the success of DevOps initiatives [6]. By identifying and focusing on factors critical to success, organizations can ensure that DevOps initiatives are in sync with overall business objectives [8, 4]. This ensures that key factors contributing to success receive the necessary attention and resources, which increases the likelihood of successful DevOps adoption, improves collaboration, and delivers high-quality software with greater efficiency [7, 4, 8].

Insights into CSFs provide a basis for continuous improvement. Regularly assessing and refining the critical success factors allows organizations to adapt and evolve their DevOps practices to meet changing business needs and technological advancements. Understanding CSFs related to cultural aspects leads to initiatives that promote a DevOps-friendly work environment, which is crucial for fostering a collaborative and innovative culture [7, 4, 8, 12, 13, 1, 3]. CSFs may highlight the importance of automation and effective tooling, which organizations should leverage insights to invest in the right tools and technologies that align with DevOps principles, enhancing efficiency and reducing manual interventions in the software delivery pipeline. CSFs often involve defining relevant metrics for measuring the success of DevOps practices. Insights into these metrics enable organizations to monitor performance, identify bottlenecks, and continuously optimize their processes [7, 4, 8, 5, 6, 9].

4.2 Implications for academia

Academic researchers play a pivotal role in conducting interdisciplinary research in DevOps, as their work is crucial for the evolution of this field [32]. However, there is a need for more research due to the education system's adaptability to evolving market demands [32]. Enhancing industrial and market relevance skills and emphasizing efficient work methodologies, such as automation tasks and agile and DevOps practices, are essential knowledge to learn. Knowing empirical research on the impact of DevOps on organizational performance and the challenges during implementation is critical [14]. Updating curricula to include DevOps practices ensures students are equipped with industry-aligned skills. Researchers must conduct various research, including agile methodologies and dynamics, and study DevOps practices. Additionally, exploring the impact of automation and efficient collaboration on productivity is crucial [14].

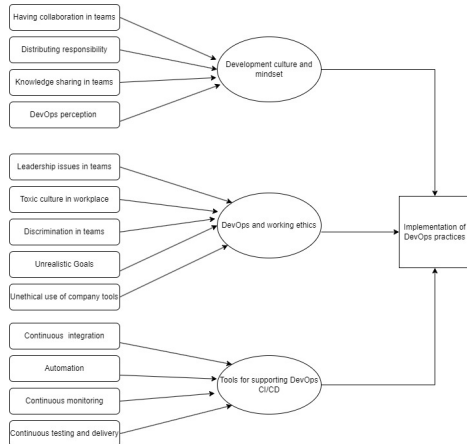


Fig. 1: Coding process for DevOps Implementation practices

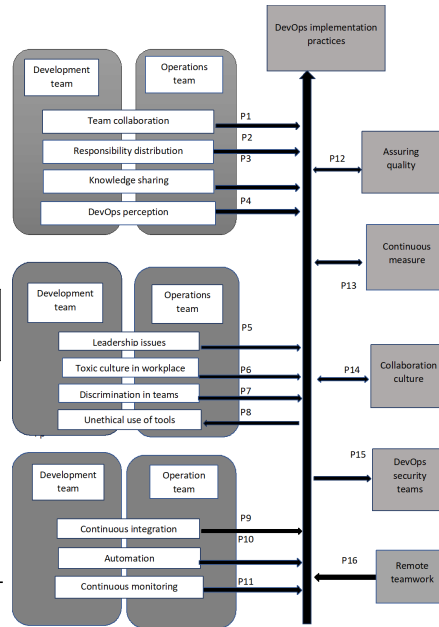


Fig. 2: Theory for DevOps critical success factors

4.3 Implications for practice

Professionals’ perspectives can be applied to DevOps practices to generate insights and strategies that are directly applicable to real-world contexts. In Table 2, we have outlined the practical guidelines recommended by industry professionals for optimizing the performance of Development and Operations practices within DevOps teams.

5 Related work

5.1 DevOps practices and success factors

Customers now have higher demands and expectations, and they expect software applications to be of higher quality and delivered in a quicker time [4]. Many different methods are utilized by companies for software development, with agile practices being one of the most widely used. However, the issue with agile practices is that they tend to only cater to the development side, leaving out the operations team. In order to have a complete software process, it is necessary to have both the development and operations teams working together [27]. In order for a software development project to generate business value, it is imperative that the software be deployed to the production environment. Without such deployment, even if the software development process was completed in an

Table 3: Practical guidelines for DevOps teams

| Practical guidelines | Recommendations for teams |
|---|--|
| Facilitate the exchange of knowledge and promote collaborative learning. | To optimize the team’s knowledge and support system, we highly recommend implementing standards for knowledge sharing and learning opportunities among team members. This involves adapting the necessary knowledge for plan, building, and run activities associated with specific products. |
| Collaborate and coordinate with teams from different departments to achieve common goals. | One effective strategy is to combine teams from different departments to take full responsibility for the entire delivery process of one or more IT products. To ensure strong teamwork and accountability, it is recommended that the team’s approach be product-focused rather than project-focused. This approach will help to create a sense of coherence and social responsibility within the team. |
| Teams must take swift action to address toxic cultures and discrimination and ethical concerns. | The team’s working culture should be professional, friendly, and free from discrimination and unethical behavior as it has a significant impact on team productivity. |
| Teams must enable the identification of critical success factors specific to the DevOps implementation within a given organization. | Organizations can focus their efforts on factors empirically linked to success, potentially accelerating the adoption and integration of DevOps practices. |
| Teams should customize training and skills development. | Training programs and skill development initiatives can be customized to address the specific needs identified within team practices by ensuring a more targeted and effective approach to process development |
| Continuous improvement based on real-world experiences. | Organizations can adapt their DevOps practices in real-time, responding to changing needs and circumstances with insights derived from an ongoing practical approaches. |

expeditious manner, the project will fail to yield any tangible benefits for the business. It is, therefore, crucial for organizations to ensure that their software development efforts culminate in the successful deployment of the software to production [27, 7, 6].

DevOps practices can improve software performance, collaboration, integration, and communication [11, 7, 4]. There is no specific definition for DevOps, though many concepts surround it [28]. The purpose of DevOps is to bridge the gap between teams and enable efficient software development [51]. DevOps merges development and operations teams and aims to deliver software features faster to customers. Although development teams deliver features more frequently, operations teams struggle to implement them rapidly [51]. After conducting a thorough content analysis, it has been found that DevOps comprises eight key components. DevOps is a development methodology that aims to bridge the gap between Development and Operations teams. The use of DevOps can facilitate communication and collaboration, enable continuous integration and delivery with automated deployment, as well as ensure quality assurance [28].

Leidecker and Bruno [34] stated that *critical success factors* The factors that require proper sustainability, maintenance, or management for a firm to achieve significant success are known as Critical Success Factors. These factors have unique characteristics, conditions, or variables, and they have a positive impact on the

firm in a particular industry. A study conducted by Ram et al. [40] suggested that we can name the factors as critical success factors if the factor is addressed in a satisfactory manner that increases the performance of the organization. They also argued that when CSFs are in practice for achieving organizational performance improvements, successful implementation may influence the relationship between CSF and improvements in organizational performance [40, 8, 4, 6, 9, 5].

Literature also suggests that team and process efficiency have some impacts on DevOps success [18]. Smite et al [44] It is suggested that team efficiency suffers due to a reluctance to ask superiors questions and a lack of understanding of the facts. The study suggests that offshore DevOps teams may be hesitant to express criticism in order to seek guidance from their superiors. According to Wiedenmann et al. [50], Cross-functional teams and teams with DevOps skills have a significant impact on organizational success. Tsanos et al. [48]. and Kolfshoten [30] claimed that mutuality, trust, commitment, and mutual respect among teams impact the team efficiency for DevOps practices.

5.2 Limitations

In this section, we discuss the threats to the validity of this study in the context of qualitative research [33, 31, 24]. Transferability. The concept of transferability is essential to determine how applicable our findings are to other settings. To achieve this, we conducted semi-structured interviews with 16 participants from LinkedIn, representing a range of backgrounds. Our research reveals informative and insightful outcomes. However, we must acknowledge that the saturation levels may have differed if the interviews had been conducted in a different sequence.

Credibility. Ensuring the research results were accurate and based on the original data was a priority. To achieve this, we took specific steps to ensure the research was credible. These included reviewing all research steps and discussing the findings several times among the authors to reduce any bias that may have been present in the study or interviews. We also recommend that other researchers validate the themes we have identified in the future, further strengthening the credibility of our findings.

Confirmability refers to the degree to which other researchers can check the findings. We do not have the participants' permission to share the transcripts of the interviews. We identified each success factor and its implications along with professionals' perspectives by quoting participants as far as possible [21].

6 Conclusion

This paper presents a thorough thematic analysis of interviews (N=16) conducted with DevOps professionals to gain valuable insights into the critical factors that impact organizational performance. One of the paper's contributions provides an enhanced definition of DevOps success factor identification for organizational software automation practices. Additionally, the detailed description of DevOps adoption and implementation presented in this paper has shown that fostering

a (1) collaborative organizational culture, (2) Integrating DevOps automation, (3) cultivating strong work ethics, (4) addressing challenges related to remote teamwork, and (5) establishing dedicated DevOps security teams are essential factors for better organizational practices. The study provides practical recommendations for organizations to improve their DevOps practices, bridging theory with actionable guidance for practitioners.

References

1. Adolph, S., Hall, W., Kruchten, P.: Using grounded theory to study the experience of software development. *Empirical Software Engineering* **16**, 487–513 (2011)
2. Al-Zahrani, S., Fakieh, B.: How devops practices support digital transformation. *International Journal of Advanced Trends in Computer Science and Engineering* **9**(3), 2780–2788 (2020)
3. Almeida, F., Simões, J., Lopes, S.: Exploring the benefits of combining devops and agile. *Future Internet* **14**(2), 63 (2022)
4. Azad, N.: Understanding devops critical success factors and organizational practices. In: 2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB). pp. 83–90. IEEE (2022)
5. Azad, N.: The impact of devops critical success factors and organizational practices (2023)
6. Azad, N., Hyrynsalmi, S.: What are critical success factors of devops projects? a systematic literature review. In: Wang, X., Martini, A., Nguyen-Duc, A., Stray, V. (eds.) *Software Business. 12th International Conference, ICSOB 2021, Drammen, Norway, December 2–3, 2021, Proceedings*. pp. 221–237. Springer (2021)
7. Azad, N., Hyrynsalmi, S.: Devops challenges in organizations: Through professional lens. In: *Software Business: 13th International Conference, ICSOB 2022, Bolzano, Italy, November 8–11, 2022, Proceedings*. pp. 260–277. Springer (2022)
8. Azad, N., Hyrynsalmi, S.: Devops critical succes factors—a systematic literature review. *Information and Software Technology* p. 107150 (2023)
9. Azad, N., Hyrynsalmi, S., Mäntymäki, M.: Understanding devops critical success factors: Insights from professionals. In: *Conference on e-Business, e-Services and e-Society*. pp. 78–90. Springer (2023)
10. Bamigbala, T.: *Data ethics and non-compliance challenges in devops* (2023)
11. Bass, L., Weber, I., Zhu, L.: *DevOps: A software architect’s perspective*. Addison-Wesley Professional (2015)
12. Ben Mesmia, W., Escheikh, M., Barkaoui, K.: Devops workflow verification and duration prediction using non-markovian stochastic petri nets. *Journal of Software: Evolution and Process* **33**(3), e2329 (2021)
13. Bezemer, C.P., Eismann, S., Ferme, V., Grohmann, J., Heinrich, R., Jamshidi, P., Shang, W., van Hoorn, A., Villavicencio, M., Walter, J., et al.: How is performance addressed in devops? In: *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*. pp. 45–50 (2019)
14. Bobrov, E., Bucchiarone, A., Capozucca, A., Guelfi, N., Mazzara, M., Masyagin, S.: Teaching devops in academia and industry: reflections and vision. In: *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment: Second International Workshop, DEVOPS 2019, Château de Villebrumier, France, May 6–8, 2019, Revised Selected Papers 2*. pp. 1–14. Springer (2020)

15. Bosch, J.: Software product lines: organizational alternatives. In: Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001. pp. 91–100. IEEE (2001)
16. Braun, V., Clarke, V.: Using thematic analysis in psychology. *Qualitative research in psychology* **3**(2), 77–101 (2006)
17. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: 2011 international symposium on empirical software engineering and measurement. pp. 275–284. IEEE (2011)
18. Díaz, J., Almaraz, R., Pérez, J., Garbajosa, J.: Devops in practice: an exploratory case study. In: Proceedings of the 19th International Conference on Agile Software Development: Companion. pp. 1–3 (2018)
19. Díaz, J., Pérez, J., Alves, I., Kon, F., Leite, L., Meirelles, P., Rocha, C.: Harmonizing devops taxonomies—a grounded theory study. *Journal of Systems and Software* **208**, 111908 (2024)
20. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software* **119**, 87–108 (2016)
21. Fernandes, M., Ferino, S., Fernandes, A., Kulesza, U., Aranha, E., Treude, C.: Devops education: an interview study of challenges and recommendations. In: Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training. pp. 90–101 (2022)
22. Ford, D., Storey, M.A., Zimmermann, T., Bird, C., Jaffe, S., Maddila, C., Butler, J.L., Houck, B., Nagappan, N.: A tale of two cities: Software developers working from home during the covid-19 pandemic. *ACM Transactions on Software Engineering and Methodology (TOSEM)* **31**(2), 1–37 (2021)
23. Gall, M., Pigni, F.: Taking devops mainstream: a critical review and conceptual framework. *European Journal of Information Systems* **31**(5), 548–567 (2022)
24. Guba, E.G.: Criteria for assessing the trustworthiness of naturalistic inquiries. *Ectj* **29**(2), 75–91 (1981)
25. Hamunen, J., et al.: Challenges in adopting a devops approach to software development and operations (2016)
26. Horton, J., Macve, R., Struyven, G.: Qualitative research: experiences in using semi-structured interviews. In: *The real life guide to accounting research*, pp. 339–357. Elsevier (2004)
27. Huttermann, M.: Devops for developers: Integrate development and operations. *The Agile Way* (2012)
28. Jabbari, R., bin Ali, N., Petersen, K., Tanveer, B.: What is devops? a systematic mapping study on definitions and practices. In: Proceedings of the Scientific Workshop Proceedings of XP2016. pp. 1–11 (2016)
29. Kallio, H., Pietilä, A.M., Johnson, M., Kangasniemi, M.: Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of advanced nursing* **72**(12), 2954–2965 (2016)
30. Kolfschoten, G.L., de Vreede, G.J., Briggs, R.O., Sol, H.G.: Collaboration ‘engineerability’. *Group Decision and Negotiation* **19**(3), 301–321 (2010)
31. Korstjens, I., Moser, A.: Series: Practical guidance to qualitative research. part 4: Trustworthiness and publishing. *European Journal of General Practice* **24**(1), 120–124 (2018)
32. Kuusinen, K., Albertsen, S.: Industry-academy collaboration in teaching devops and continuous delivery to software engineering students: towards improved industrial relevance in higher education. In: 2019 IEEE/ACM 41st International Conference on

- Software Engineering: Software Engineering Education and Training (ICSE-SEET). pp. 23–27. IEEE (2019)
33. Larios Vargas, E., Aniche, M., Treude, C., Bruntink, M., Gousios, G.: Selecting third-party libraries: The practitioners' perspective. In: Proceedings of the 28th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering. pp. 245–256 (2020)
 34. Leidecker, J.K., Bruno, A.V.: Identifying and using critical success factors. Long range planning **17**(1), 23–32 (1984)
 35. Leite, L., Pinto, G., Kon, F., Meirelles, P.: The organization of software teams in the quest for continuous delivery: A grounded theory approach. Information and Software Technology **139**, 106672 (2021)
 36. Luz, W.P., Pinto, G., Bonifácio, R.: Building a collaborative culture: a grounded theory of well succeeded devops adoption in practice. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1–10 (2018)
 37. Meyer, A.N., Fritz, T., Murphy, G.C., Zimmermann, T.: Software developers' perceptions of productivity. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. pp. 19–29 (2014)
 38. Mohan, V., Othmane, L.B.: Secdevops: Is it a marketing buzzword?-mapping research on security in devops. In: 2016 11th international conference on availability, reliability and security (ARES). pp. 542–547. IEEE (2016)
 39. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. Information and Software Technology **56**(10), 1200–1218 (2014)
 40. Ram, J., Corkindale, D., Wu, M.L.: Implementation critical success factors (csfs) for erp: Do they contribute to implementation success and post-implementation performance? International journal of production economics **144**(1), 157–174 (2013)
 41. Shahin, M., Babar, M.A., Zahedi, M., Zhu, L.: Beyond continuous delivery: an empirical investigation of continuous deployment challenges. In: 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). pp. 111–120. IEEE (2017)
 42. Sjøberg, D.I., Dybå, T., Anda, B.C., Hannay, J.E.: Building theories in software engineering. Guide to advanced empirical software engineering pp. 312–336 (2008)
 43. Skenderi, M., Luma-Osmami, S., Imeri, F.: Ethics in devops, the attitude of programmers towards it. Journal of Natural Sciences and Mathematics of UT **5**(9-10), 69–85 (2020)
 44. Šmite, D., Moe, N.B., Gonzalez-Huerta, J.: Overcoming cultural barriers to being agile in distributed teams. Information and Software Technology **138**, 106612 (2021)
 45. Smith, J.A.: Semi structured interviewing and qualitative analysis (1995)
 46. Smolander, K., Rossi, M., Purao, S.: Software architectures: blueprint, literature, language or decision? European Journal of Information Systems **17**, 575–588 (2008)
 47. Strauss, A., Corbin, J.: Basics of qualitative research. Sage publications (1990)
 48. Tsanos, C.S., Zografos, K.G., Harrison, A.: Developing a conceptual model for examining the supply chain relationships between behavioural antecedents of collaboration, integration and performance. The International Journal of Logistics Management (2014)
 49. Ur Rahman, A.A., Williams, L.: Security practices in devops. In: Proceedings of the Symposium and Bootcamp on the Science of Security. pp. 109–111 (2016)
 50. Wiedemann, A., Wiesche, M., Krcmar, H.: Integrating development and operations in cross-functional teams-toward a devops competency model. In: Proceedings of the 2019 on Computers and People Research Conference. pp. 14–19 (2019)

51. Zarour, M., Alhammad, N., Alenezi, M., Alsarayrah, K.: A research on devops maturity models. *Int. J. Recent Technol. Eng* **8**(3), 4854–4862 (2019)