frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 01 September 2025
DOI 10.3389/fcomp.2025.1554299

OPEN ACCESS

EDITED BY
Martin Shepperd,
Brunel University London, United Kingdom

REVIEWED BY
Sabina Rossi,
Ca' Foscari University of Venice, Italy
Lindung Parningotan Manik,
National Research and Innovation Agency (BRIN), Indonesia

*CORRESPONDENCE
Halil Ergun Korkmaz
✉ halilergun.korkmaz@gmail.com

RECEIVED 01 January 2025
ACCEPTED 07 August 2025
PUBLISHED 01 September 2025

CITATION
Korkmaz HE and Aydin MN (2025) An empirical study on performance comparisons of different types of DevOps team formations. *Front. Comput. Sci.* 7:1554299.
doi: 10.3389/fcomp.2025.1554299

An empirical study on performance comparisons of different types of DevOps team formations

Halil Ergun Korkmaz[1]* and Mehmet Nafiz Aydin[2]

[1]Management Information Systems, Kadir Has University, Istanbul, Türkiye, [2]Management Information Systems, Bogazici University, Istanbul, Türkiye

Introduction: Despite all the efforts to successfully implement DevOps practices, principles, and cultural change, there is still a lack of understanding on how DevOps team structure formation and performance differences are related. The lack of a ground truth for DevOps team structure formation and performance has become a persistent and relevant problem for companies and researchers.
Methods: In this study, we propose a framework for DevOps team Formation–Performance and conduct a survey to examine the relationships between team formations and performance with the five metrics we identified, two of which are novel. We conducted an empirical study using a survey to gather data. We employed targeted outreach on a social media platform along via a snowball sampling and sent 380 messages to DevOps professionals worldwide. This approach resulted in 122 positive responses and 105 completed surveys, achieving a 69.7% response rate from those who agreed to participate.
Results: The research shows that implementing the DevOps methodology enhances team efficiency across various team structures, with the sole exception of "Separate Development and Operation teams with limited collaboration". Moreover, the study reveals that all teams experienced improvements in Repair/Recovery performance metric following DevOps adoption. Notably, the "Separate Development and Operation teams with high collaboration" formation emerged as the top performer in the key metrics, including Deployment Frequency, Number of Incidents, and Number of Failures/Service Interruptions. The analysis further indicates that different DevOps organizational formations do not significantly impact Lead Time, Repair/Recovery, and Number of Failures/Service Interruptions in terms of goal achievement. However, a statistically significant disparity was observed between "Separate Development and Operation teams with high collaboration" and "A single team formation" regarding the Deployment Frequency goal achievement percentage.
Discussion: The analysis confirms that DevOps adoption improves performance across most team formations, with the exception of "Separate Development and Operation teams with limited collaboration" (TeamType1), which shows significant improvement only in Mean Time to Recovery (MTTR). Standardized effect size calculations (Cohen's d) reveal that TeamType2 ("Separate Development and Operation teams with high collaboration") consistently achieves large effects in Deployment Frequency (DF), Number of Incidents (NoI), and Number of Failures/Service Interruptions (NoF/NoSI), while TeamType3 shows strong results for Lead Time (LT) and NoF/NoSI. MTTR improvements are large across all formations, with TeamType4 performing best in this metric. These findings suggest that collaboration intensity is a critical determinant of performance gains. While team formation type does not significantly influence LT, MTTR, or NoF/NoSI goal achievement, DF goal achievement is significantly higher for TeamType2 compared to TeamType4, highlighting the potential competitive advantage of high-collaboration structures.

KEYWORDS

DevOps, team structure, performance comparison, DevOps taxonomy, DevOps formations

1 Introduction

Delivering continuously fast and reliable products and services to customers requires high-level coordination, cooperation, and information flow between the development and IT operation teams within the technology departments. One can argue that there is a gap and disconnection between development and IT operations for continuous delivery and deployment, whereas continuous improvement exists independently of each other. This issue was first addressed by P. Debois at the 2008 Agile Conference in Toronto, known today as DevOps (Debois, 2008).

The consensus about the very idea of DevOps is that it is a set of practices that maximizes collaboration between development (Dev) and IT operations (Ops) to reduce software development and delivery time while increasing quality (Loukides, 2012). It also shapes the organizational structure of Development and IT operations for successful delivery and deployment (Lwakatare et al., 2016b). Another view of DevOps is that it is an extension of Agile software development. It maintains culture, automation, and measurement as principles of agile methods, such as XP and Lean, as well as flow, feedback, continuous learning, and implementation (Kim et al., 2016). Despite all endeavors for achieving successful DevOps, there is still a lack of understanding of the extent to which the formation of DevOps' team structures and performance differences are correlated.

While DevOps is highly effective for companies to react to customer demands with fast and reliable software development and delivery, they still need to address coordination and collaboration among different teams in IT organizations. Practitioners generally accept that platforms and automation tools are essential for DevOps adoption and make DevOps teams more efficient (Leite et al., 2019). However, ineffective DevOps team formation results in inadequate coordination and collaboration between teams and inefficient usage of these platforms and automation tools.

As a result, customer dissatisfaction and disappointment emerge, potentially leading to both reputational damage and financial losses in the marketplace. In response to these risks, companies increasingly invest in the development of their IT team structures and make substantial payments to solution providers in pursuit of more efficient organizational and team configurations. Identifying effective DevOps team formations, if they exist, can not only help organizations mitigate financial losses but also prevent reputational harm. For these reasons, this topic has drawn significant attention among both researchers and practitioners.

As there is no ground truth for DevOps' team structure formation from a performance perspective, it has become a persistent and relevant issue for companies and researchers (Leite et al., 2023; Lopez-Fernandez et al., 2021). They are looking for the formation of DevOps' team structures and performance differences. Thus, we focused on this topic using the following research questions:

- RQ1: Does adopting DevOps practices create a performance difference for all types of DevOps Teams?
- RQ2: Considering both pre-DevOps and post-DevOps performance values, which DevOps team formation shows greater performance improvement compared to others?
- RQ3: Which DevOps team type is better than others in terms of performance goals achievements?

To address these research questions, we adopted three existing performance metrics (Lopez-Fernandez et al., 2021; DORA, n.d.) which are Lead Time (LT), Deployment Frequency (DF), and Mean Time To Repair/Recovery (MTTR). In addition to these three metrics, we propose two novel metrics which are Number of Incidents (NoI) and Number of Failures/Service Interruptions (NoF/NoSI) metrics as articulated further later on, these metrics are concerned with service quality and reliability, which is a missing DevOps performance metric in the existing studies.

We conducted a survey by adopting a quantitative approach. To measure the performance difference of DevOps adoption Pre-DevOps and post-DevOps values of the five performance metrics were requested. Participants were also asked to provide the percentage of goal achievement for each performance metric except NoI. The aim is to compare the performances of the DevOps groups and determine which group performed better.

The paper is structured as follows: In Section 2, we describe the relevant works, and in Section 3, we describe the present research methodology, performance metrics, and a framework. We present the statistical methods for data analysis and the results in Section 4. Finally, we provide the research limitations, discussion, and conclusions in Sections 5 and 6, respectively.

2 Relevant works

Despite the many studies on DevOps in general, there are a limited number of studies focusing on the relationships between Formation-Performance of DevOps as summarized by Table 1. Muñoz et al. (2021) pointed out that there are three main problems in implementing DevOps practices and summarized them as lack of a defined process (Muñoz et al., 2019), lack of guidance (Hüttermann, 2012), and lack of knowledge and experience (Senapathi et al., 2018).

In "Model Checking Adaptive Multilevel Service Compositions," Rossi (2012) introduces a logic-based method for verifying security and correctness in service compositions. This approach aligns with DevOps principles by promoting continuous verification and integration, ensuring that adaptive services maintain compliance throughout their lifecycle.

In their study, Díaz et al. (2021) investigate the motivations behind companies adopting a DevOps culture. Through an exploratory multiple case study involving 30 multinational software-intensive companies, they conducted in-depth, semi-structured interviews with key stakeholders. The research identifies common challenges organizations face in accelerating software delivery and highlights the primary drivers for embracing DevOps practices. Findings reveal

TABLE 1 Summary of relevant research.

Representative study	Focus of research	Research approach	Key research outcomes	Limitations
Muñoz et al. (2021)	Using a platform based on the basic profile of ISO/IEC 29110 to reinforce DevOps environments	Systematic Literature Review	A guidance, and a web platform to implement and/or to reinforce a DevOps approach	A single case study with one small Scrum-based organization, which restricts broader applicability, and focus on diagnostic insights rather than implementation results. It relies on user input accuracy, targets only the ISO/IEC 29110 Basic profile, and has not been validated across diverse organizational contexts.
Lwakatare et al. (2016a)	An exploratory study of DevOps: extending the dimensions of DevOps with practices	An exploratory case study	A scientific definition of DevOps and patterns of DevOps practices to help identify and adopt the phenomenon.	Exploratory, non-empirical nature and dependence on limited qualitative input. Lack of actionable adoption guidance, broad validation, and does not address how definitions and practices change across sectors.
Shahin et al. (2017)	Adopting continuous delivery and deployment: impacts on team structures, collaboration and responsibilities	A mixed-method (using both qualitative and quantitative research methods for data collection and analysis) empirical study	Identified four key patterns for structuring Dev and Ops teams to effectively initiate CD practices	Findings are based on limited representativeness, self-reported data, and qualitative bias, and lack outcome validation over time, resulting in restricted generalizability.
Lopez-Fernandez et al. (2021)	DevOps team structures: characterization and implications	Based on the constructivism model as underlying philosophy. Collecting rich qualitative data and apply Grounded Theory.	A taxonomy for DevOps team structure patterns. Reveals four team structures that companies have been adopting	No causal or predictive claims, focuses on taxonomy building. Possible interpretation of subjectivity despite inter-coder reliability. Only software-intensive multinational companies are included.
Leite et al. (2021)	Type of Organizational structures adopted by software-producing organizations. The properties of that structures and the performance comparison	Collecting and analyzing data from semi-structured interviews with IT professionals, following Grounded Theory guidelines	Identified four common organizational structures. Also observed that some companies are transitioning between these structures.	Limited by a small and regionally concentrated sample—mainly from Brazil—which may not reflect broader industry or cultural contexts. Additionally, while Grounded Theory offers explanatory insights, it does not establish causality or enable prediction.
Leite et al. (2023)	The rationale behind the selection of different organizational structures, and management of its drawbacks	Interviewing selected IT professionals from different countries and analyze the conversations through a Grounded Theory process	Identified conditions, causes, reasons to avoid, consequences, and contingencies related to each discovered structure and offer a theory to explain organizational structures for development and infrastructure professionals.	Does not offer probabilistic forecasts in terms of dependent and independent variables. Certain factors (including the anonymity of interviewees and the inevitable subjective nature of analysis) render the study not entirely reproducible. Due to cost considerations, the review process has limitations.
Díaz et al. (2024)	The study aims to harmonize existing knowledge on DevOps team structures to provide a unified understanding of organizational structures and characteristics in DevOps adoption.	Empirical studies conducted to build a theory. Grounded Theory methodology employed. Existing theories and representations of team taxonomies are analyzed	A theory with 28 propositions mapping properties to constructs of the DevOps Team Taxonomies. 34 testable hypotheses generated. 11 were thoroughly tested. Testing proved the framework's effectiveness. The constructs enhanced the initial framework.	The testing phase thoroughly examined only 11 out of the 34 generated hypotheses. This suggests that further empirical validation is needed to confirm the remaining hypotheses and strengthen the theory's generalizability.

(Continued)

10.3389/fcomp.2025.1554299

TABLE 1 (Continued)

Representative study	Focus of research	Research approach	Key research outcomes	Limitations
Díaz et al. (2021)	The study aims to elucidate the real-world motivations driving companies to adopt DevOps practices and the outcomes they anticipate from such a cultural shift. It seeks to provide both practitioners and researchers with a comprehensive understanding of the challenges organizations face in accelerating software delivery and the primary factors prompting the transition to DevOps.	An exploratory multiple case study is conducted. In-depth, semi-structured interviews, supplemented by industrial workshops and on-site observations used as data collection methods. To enhance the reliability of their qualitative findings and mitigate potential author bias, an inter-coder agreement analysis was performed.	Identified several core issues and expected benefits that motivate companies to embrace DevOps culture. A series of patterns and anti-patterns related to the motivations behind DevOps adoption unveiled. Offering valuable insights for organizations considering DevOps transformation.	Focused exclusively on multinational software-intensive companies, which may limit the generalizability of the findings The reliance on qualitative data, though rich in context, may introduce subjectivity. The data reflects the current state of DevOps adoption, and subject to change by the evolution of technology and organizational practices.
Zhou et al. (2022)	To explore how software teams across different companies organize themselves to implement DevOps and microservices, and to examine the associated benefits and challenges.	The research approach used in this study is qualitative and ethnographic. The authors conducted a cross-company ethnographic study involving in-depth interviews and participant observations across multiple organizations.	The study highlights that teams are typically small, cross-functional, and autonomous, enabling faster delivery and better collaboration. It also identifies key challenges such as increased complexity, communication overhead, and tooling fragmentation. The findings offer practical insights into both the organizational benefits and the operational difficulties associated with DevOps and microservices adoption.	Involved only three companies, which limits the generalizability of its findings. Remote observations during the COVID-19 pandemic may have further reduced the depth of ethnographic insights. Therefore, the results should be interpreted with caution in broader contexts.
Gomes et al. (2022)	The research focuses identification, definition, and evaluation of Key Performance Indicators (KPIs) tailored specifically for assessing the performance and effectiveness of DevOps teams. The study aims to establish a comprehensive framework of measurable metrics that can be used to quantitatively evaluate various aspects of DevOps team operations, thereby facilitating improved monitoring, management, and optimization of DevOps practices in organizational settings.	The researchers utilize a systematic literature review to identify and analyze key performance indicators (KPIs) for evaluating DevOps teams. Their study consolidates existing research to highlight 13 important KPIs, especially those related to testing quality and outcomes, providing a comprehensive framework for assessing DevOps team effectiveness based on established metrics.	The researchers develop and validate a comprehensive set of Key Performance Indicators (KPIs) specifically designed to evaluate the performance of DevOps teams. Their framework categorizes KPIs across different dimensions of DevOps practices, providing organizations with practical tools to measure team effectiveness and delivery outcomes. The study emphasizes the applicability of these KPIs in real-world settings, supporting continuous improvement and informed decision-making within DevOps environments.	Due to the relatively small number of existing works focused specifically on DevOps team performance measurement, which may affect the generalizability of their proposed KPIs. Additionally, the study does not address practical challenges in implementing these KPIs within organizations, such as data collection and integration issues. Future research is needed to validate these KPIs in real-world settings and explore obstacles to their adoption.

(Continued)

10.3389/fcomp.2025.1554299

10.3389/fcomp.2025.1554299

TABLE 1 (Continued)

Representative study	Focus of research	Research approach	Key research outcomes	Limitations
Williams et al. (2025)	Identifying, defining, and evaluating key metrics and key performance indicators (KPIs) that effectively measure the success of DevOps initiatives. The study aims to establish a standardized framework for assessing DevOps performance, enabling organizations to quantitatively track progress, optimize processes, and align DevOps practices with business objectives.	The researchers adopt a quantitative, survey-based research approach to identify and validate key metrics and KPIs associated with DevOps success. By collecting data from DevOps professionals across various industries and applying statistical analyses, the study offers empirical insights into the performance indicators that most effectively reflect successful DevOps practices and outcomes.	Propose a comprehensive framework for measuring DevOps success by identifying key metrics and KPIs across technical and organizational domains. The study highlights the importance of aligning these metrics with strategic business objectives and emphasizes the need for context-specific measurement practices.	The findings may be context-specific and not easily generalizable across industries or organization sizes. Reliance on self-reported data poses a risk of bias, and the fast-changing nature of DevOps practices may impact the long-term relevance of the proposed metrics.
Plant et al. (2022)	Design and validation of a structured measurement instrument to assess the capabilities of DevOps teams. The study aims to identify key capabilities that influence DevOps team effectiveness and to develop a practical, empirically validated tool that organizations can use to evaluate and improve these capabilities within their teams.	Design Science Research (DSR) is used as a research approach. Focuses on the iterative development and evaluation of an artifact—in this case, a capability measurement instrument—intended to solve a practical problem within a specific context. The authors followed the DSR methodology to systematically design the instrument, validate it through expert feedback, and empirically test it using survey data from DevOps practitioners.	The development and validation of a reliable measurement instrument that assesses essential capabilities of DevOps teams. This tool helps organizations evaluate their teams' strengths and areas for improvement, supporting more effective DevOps implementation and continuous capability development.	Small sample size and the absence of statistical validation methods such as factor analysis or internal consistency testing. The reliance on self-assessment also raises concerns about subjectivity, particularly in smaller teams, where qualitative methods might offer deeper insights. These limitations suggest the need for broader empirical testing and refinement of the measurement tool.

patterns and anti-patterns related to DevOps adoption, offering insights to help practitioners and researchers understand the contexts and issues prompting companies to transition towards DevOps. This study aims to strengthen evidence and support practitioners in making better-informed decisions about which problems trigger a DevOps transition and the most common expected results.

Another study characterizing the DevOps phenomenon is explored by Lwakatare et al. (2016a) via a review of multivocal 'grey' literature and interviews (Leite et al., 2020; Leite et al., 2021) provide further conceptual elaborations of DevOps using a set of exemplary practices and patterns.

Zhou et al. (2022) conducted an ethnographic study to comprehensively examine the organization, benefits, and problems of software teams using DevOps and microservices in three companies with different business, size, product, customer, and degree of globalization. The researchers collected data by conducting nine interviews with software teams related to DevOps and microservices over 7 months and tested their theory by conducting a comparative analysis of the companies with this data. The study contributes to a better understanding of the benefits and problems of adopting DevOps and microservices for organizations, but is limited to reveal an extent to which different team types affect DevOps performance.

Gomes et al. (2022) focus on developing a structured set of Key Performance Indicators (KPIs) to evaluate the performance of DevOps teams. Based on a systematic review of the literature, they identified and organized KPIs into key dimensions such as operational efficiency, collaboration, automation, and continuous delivery. Their goal was to help organizations assess DevOps practices more effectively and align performance measurement with business objectives. This work provides a practical framework that can guide organizations in evaluating and improving the effectiveness of their DevOps teams, but needs to be validated in an empirical setting.

In another DevOps performance measurement study, Williams et al. (2025) examined basic DevOps metrics and Key Performance Indicators (KPIs) to measure the effectiveness, speed, and reliability of DevOps practices. In this study, the researchers examined deployment frequency, lead time for changes, mean time to repair, change failure rate, also known as DORA (n.d.) metrics, as well as KPIs such as customer satisfaction, team collaboration, and infrastructure automation levels. They suggest that using both quantitative and qualitative metrics together, DevOps performance can be aligned with the organization's business goals with a balanced approach.

Plant et al. (2022) present the development and empirical validation of a capability measurement instrument specifically designed for DevOps teams. Through a design science research methodology, the study identifies and structures key capabilities required for effective DevOps performance and translates these into measurable constructs. The resulting instrument is validated through expert evaluations and a survey administered to DevOps practitioners, demonstrating its reliability and practical relevance. This work contributes to the DevOps field by offering a systematic and evidence-based tool to assess and enhance team capabilities, thereby supporting continuous improvement and strategic alignment in DevOps environments.

Jayakody and Wijayanayake (2023), in their study, examined the Critical Success Factors (CSF) for DevOps adoption by stating the benefits and difficulties of DevOps adoption and analyzed the use of DevOps principles and principles. They used the systematic literature review method while determining the Critical Success Factors and verified the results with interviews with DevOps experts. At the end of the study, they presented a conceptual model output in which they grouped the Critical Success Factors they determined under four main headings such as collaborative culture, DevOps practices, proficient DevOps team, and metrics and measurement.

Shahin et al. (2017) highlighted in their paper that DevOps adoption helps organizations improve skills, form the right teams, and investigate organizational processes, practices, and tool support. It is emphasized that continuous Delivery/Deployment (CD) may require a new way of working and changes in team structures and responsibilities. Furthermore, it is argued that CD practices demand tighter and stronger collaboration and integration among teams and team members (Shahin et al., 2017). They conducted an empirical investigation of team organization, collaboration, and responsibility from the perspective of DevOps adoption by conducting a mixed-method study consisting of interviews and surveys with software practitioners from 19 organizations and 93 practitioners. They conclude that there are four common patterns for organizing Dev and Ops teams to effectively initiate and adopt CD practices. The participants shared that team co-location, rapid feedback, joint work, shared responsibility, collaboration tools, awareness, transparency, empowering, and engaging operations personnel increased collaboration among teams and team members in the CD adoption path. They highlighted that after DevOps adoption, their responsibilities changed, especially in terms of their skills, new CD-related solutions, and task prioritization. They also found that adopting CD enhances the relationship between responsibilities and skill sets in both negative and positive ways.

In another research study, Leite et al. (2020) focused on DevOps Team structure patterns in their paper and conducted 20 semi-structured interviews with 27 IT professionals. They used Grounded Theory (Glaser and Strauss, 1999) as a methodology for analyzing data and concluded with four organizational structures. Leite et al. (2021) mainly focus on team structures as well as types of DevOps organizational structures (Leite et al., 2023). During the interview stage, they identified the core and supplementary properties of each team structure type. Additionally, they used "Lead Time," "Deployment Frequency" and "Mean Time To Repair" as three software delivery metrics (Leite et al., 2021) to find out organizational structure differences. In previous studies, Leite et al. concluded in their study that while "Platform team" is better than other team structures for achieving continues delivery, "Siloed Departments" closer to less-than-high performance organizations. Additionally, they could not find any relation between delivery performance and "Classical DevOps" and "Cross Functional Teams" (Leite et al., 2021). In their follow-up study (Leite et al., 2023), the authors focused on the rationale behind the selection of different types of DevOps team structure and the management of drawbacks for each type. In this study, 68 IT professionals from 54 organizations were interviewed. They used "Grounded Theory" once again to build their theory. Indirectly, they similarly addressed collaboration, responsibility, and skills as constructs in their studies. They claimed that they found the rationale for different types of DevOps team structure adoption by organizations, such as company size, to overcome delivery issues, compliance with organizational standards, or resource assignment requirements. They also claimed that they found the disadvantages of each type of team structure and the solutions that are applied by the organizations for remediations.

Yet another study conducted by Lopez-Fernandez et al. (2021) focuses on DevOps team structure taxonomy. They conducted an exploratory study using semi-structured interviews with 31 multinational software-intensive companies, attending industrial workshops, and visiting organizational facilities. Similar to others, they used "Grounded Theory" (Glaser and Strauss, 1999) as a qualitative research method to explore the structure and characteristics of teams and statistical analysis to discover their implications in software delivery performance. While describing a taxonomy of team structure patterns, they used the following six constructs: collaboration frequency, product ownership sharing, autonomy, leadership management, organizational silo, and cultural silo. They also used "Lead Time," "Deployment Frequency" and "Mean Time to Repair" as three software delivery metrics same as others to find out the implications on software delivery performance. They concluded with four different types of DevOps team structures similar with Shahin et al. (2017), Leite et al. (2020), Leite et al. (2021), and Leite et al. (2023) and claim that "Full cross-functional DevOps team" performs better than "Interdepartmental Dev and Ops collaboration" and "Interdepartmental Dev-Ops team." They also found that "Boosted cross-functional DevOps team" and "Full cross-functional DevOps team" have better software delivery performance (Lopez-Fernandez et al., 2021).

Díaz et al. (2024) address the need for a unified understanding of DevOps team structures within software-producing organizations in the article "Harmonizing DevOps Taxonomies—A Grounded Theory Study." They emphasize that if companies improve their software development and deployment process, DevOps becomes essential and requires significant cultural and organizational changes. The researchers sought to enhance the understanding of the organizational structure and characteristics of teams implementing DevOps by synthesizing existing knowledge. The researchers conducted an empirical investigation to synthesize existing knowledge on team taxonomies in order to facilitate a systematic and structured DevOps adoption. The grounded theory (Glaser and Strauss, 1999) approach and Inter-Coder Agreement (ICA) were both used as research methods in the study. In addition to these two methods, the researchers analyze existing studies on DevOps teams and taxonomies to gain necessary knowledge about the subject. In conclusion, the authors presented a theory of DevOps taxonomies, asserting its nature as both substantive and analytic. Additionally, they provided a public repository containing pertinent data related to the study.

3 Research methodology

Existing studies (Shahin et al., 2017; Leite et al., 2020; Leite et al., 2021; Leite et al., 2023; Lopez-Fernandez et al., 2021; Díaz et al., 2024) employed qualitative methods together with grounded theory (Glaser and Strauss, 1999) to build and test their theories. Although these studies provide insights for developing our theoretical framework and identifying number of team formations in advance, we decided to test our theory by using quantitative methods and inferential statistical analysis by using known statistical tests such as Variance Analysis (ANOVA), *t*-test, and Chi-square. As elucidated in depth in subsequent sections, a set of questions was formulated to address the five performance metrics that constitute the focus of this study. These questions were published as a survey form via the Internet. Participants were recruited using "Snowball Effect" (Biernacki and Waldorf, 1981), "Targeted Advertising" (Thornton et al., 2016) and "Crowdsourcing via Open Calls" (Kosinski et al., 2015) methodologies to ensure expertise in DevOps profiles. Due to unknown population size, an *a priori* power analysis was conducted using G*Power 3.1.9.7 (Cohen, 1988) to estimate the required sample size for a one-way ANOVA comparing four independent groups. Assuming a large effect size ($f = 0.40$), an alpha level of 0.05, and statistical power of 0.80, the minimum required total sample size was determined to be 76 participants, with 19 individuals per group. This value served as a threshold for sample size, ensuring the study was adequately powered to detect substantial group differences (Faul et al., 2007). Although the number of participants exceeded the initially established sample size of 76 and reached a total of 104, the response count for one of the DevOps team groups (TeamType1) remained constant at 9. Consequently, the survey was concluded, and the study proceeded with data analysis and interpretation of the results.

The survey was concluded after participant enrollment reached a saturation point, subsequent to which data analysis and interpretation of findings were conducted.

3.1 Identifying metrics and framework

While we were identifying the examined metrics, we applied the predecessors' papers again and found that both Lopez-Fernandez et al. (2021) as referring to DORA (n.d.) and Leite et al. (2021) used the following software delivery performance metrics in their research to compare the different types of DevOps team structure performance: "Lead Time," "Deployment Frequency" and "Mean Time To Repair." In addition to these three software delivery performance metrics, we propose to add two further metrics "Number of Incidents" and "Number of Failures/Service Interruption" in the model to measure applications or service quality and reliability. While the "Number of Failures/Service Interruptions" gives us only service outages, the "Number of Incidents" gives us any type of service degradation. We also see that DORA (n.d.) added a new performance metric, "Change failure percentage, "to measure the impact of service degradation requiring remediation. They preferred to use a single metric for all types of service deterioration and outages. However, we suggest that using the "Number of Incidents" and "Number of Failures/Service Interruption" metrics separately will provide a detailed performance comparison opportunity. Therefore, we propose a framework as shown in Figure 1.

Regarding DevOps formations, similar with other studies (Shahin et al., 2017; Leite et al., 2020; Leite et al., 2021; Leite et al., 2023; Lopez-Fernandez et al., 2021) we identify four types of DevOps team structures, based on the level of collaboration, support model and organizational relationships. The name of the four DevOps team structures and their definitions are listed as follows:

- Separate Development and Operations Teams with Limited Collaboration (TeamType1): Teams often perceive DevOps principles primarily as the implementation of continuous

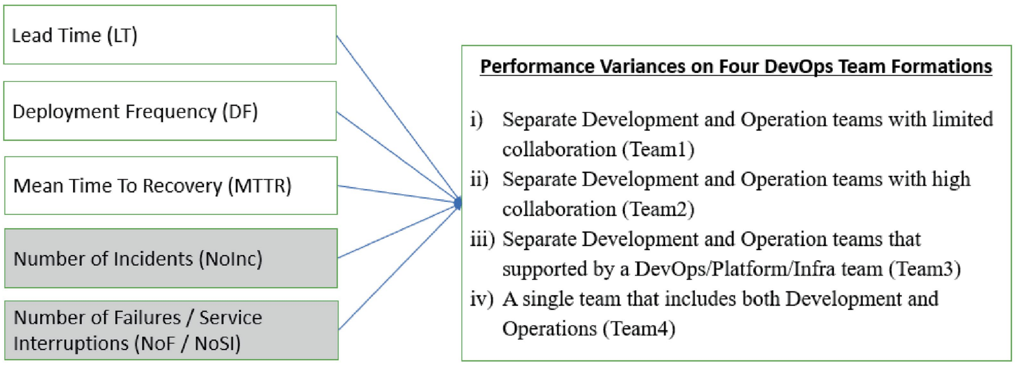

FIGURE 1
DevOps team performance metrics (the shaded metrics belong to this research).

integration tools rather than as an initiative to enhance collaboration. Consequently, communication and collaboration between teams remain minimal. Developers and operations personnel engage in infrequent collaboration, occurring on a monthly or, at most, a weekly basis. Each group primarily focuses on its own tasks, leading to a mere transfer of responsibilities rather than active cooperation. Moreover, these teams lack autonomy, as responsibilities are delegated, and dependencies on other teams persist. This type of team structure has been identified in other studies under different terminologies, such as "Siloed Departments" (Leite et al., 2023) and "Interdepartmental Dev and Ops Collaboration" (Lopez-Fernandez et al., 2021).

- Separate Development and Operations Teams with High Collaboration (TeamType2): In this team structure, there is significant collaboration between developers and infrastructure teams. While Development and Operations function as a stable team, they may still belong to distinct departments. As a result, they might have a shared product manager, but often, they report to separate department managers with differing objectives. Task allocation remains clearly delineated between the Development and Operations teams. While organizational silos have been addressed, cultural silos continue to persist. However, collaboration extends beyond the mere transfer of tasks, as team members begin to share product ownership and gain increased autonomy. This team structure has been described in other studies using different terminologies, such as "A Classical DevOps Structure" (Leite et al., 2023) and "Interdepartmental Dev-Ops Team" (Lopez-Fernandez et al., 2021). An illustrative example of this team model is Google's Site Reliability Engineering (SRE) approach (Beyer et al., 2016), wherein developers and SRE teams operate independently yet collaborate extensively to achieve shared objectives.
- Separate Development and Operations Teams Supported by a DevOps/Platform/Infrastructure Team (TeamType3): In this model, Development and Operations teams remain distinct but receive support from DevOps experts affiliated with a DevOps Center of Excellence or platform/infrastructure teams, particularly in large-scale enterprises. These supporting teams provide highly automated infrastructure services, ensuring the implementation of DevOps best practices. While Development and Operations teams continue to function separately, they collaborate closely on a daily basis and possess a moderate to high degree of autonomy. This team structure has been referred to in other studies as a "Platform Team" (Leite et al., 2023) and a "Boosted Cross-Functional DevOps Team" (Lopez-Fernandez et al., 2021).
- A Single Team Integrating Both Development and Operations (TeamType4): This model consists of a self-sufficient team that encompasses both development and operations expertise, enabling end-to-end product development. These teams operate under a single manager and collectively assume ownership of the product or service they deliver. Working closely together on a daily basis, they exhibit a medium-to-high level of autonomy. This team structure has been described in prior research as a "Cross-Functional Team" (Leite et al., 2023) and a "Full Cross-Functional DevOps Team" (Lopez-Fernandez et al., 2021). As highlighted by Leite et al. (2023) in their paper, this approach aligns with Amazon's principle of "You built it, you run it" (Gray, 2006) and the concept of "autonomous squads" employed at Spotify (Kniberg, 2014).

We also added the "Other" option to the relevant survey questions if the participants wanted to add different options. There is only one participant used the "Other" option with adding "We have almost every kind of these options" comment. Therefore, we excluded that participant from the analysis as an outlier. We intentionally avoided naming DevOps team structure types because naming conventions can be variable and named differently by company. Instead, we define DevOps team structures based on the level of collaboration, support model and organizational relationships.

3.2 Data gathering

We used similar questions with the predecessors who were studied on the same subject such as Lopez-Fernandez et al. (2021), Leite et al. (2020), Leite et al. (2021), Leite et al. (2023), and Shahin et al. (2017). We published the survey through Google Forms[1] and all questions

1 .https://drive.google.com/file/d/1R27WeYrOuUcawXG9NiEvOMf0DpTNsPq_/view?usp=drive_link

were marked as mandatory. In addition to the 9 demographic questions, 14 questions were related performance metrics. In the survey, we had 22 multiple-choice and 1 multiple-select questions. We also added an "Other" option to some questions in case participants wanted to add their input. The survey was completed anonymously, and we did not ask for any personal or company-related information from participants except for an email address that was not mandatory and asked only if they wanted to obtain the survey results.

The process of identifying and accessing survey participants presents significant challenges. The use of social media platforms for survey research offers methodological advantages and innovative approaches such as "snowball effect" (Biernacki and Waldorf, 1981), "targeted advertising" (Thornton et al., 2016), "random sampling within groups" (Brickman Bhutta, 2012), "network-based sampling" (Heckathorn, 1997) and "Crowdsourcing via Open Calls" (Kosinski et al., 2015). These methods facilitate the process of reaching the target audience, defining participant demographics, and enhancing survey participation rates.

Initially, we decided to use "snowball effect" (Biernacki and Waldorf, 1981) method and shared the survey with our peers' network who are DevOps experts and work for global technology companies, telecommunications service providers, and financial institutions. However, after 30 days publishing the survey, the total number of participants was 19. We decided to use "Crowdsourcing via Open Calls" (Kosinski et al., 2015) method. By using this method, we post open calls for survey on social media such as LinkedIn and Facebook DevOps groups. Similar to Shahin et al. (2017), this method also failed, and we did not obtain any additional responses from those social media groups. Finally, we tried yet another method which is "targeted advertising" (Thornton et al., 2016) and searched DevOps profiles from different countries globally on LinkedIn and sent connection requests for each. Once they accepted, we sent a message, explained the research goals, and asked them to participate in the survey. We sent 380 messages and received 122 positive responses (32.1% success rate in total messages) that they would participate in, but we only obtained 105 (69.7% success rate in total response) participants in the survey[2].

After excluding the outlier record who selected the "Other" option for DevOps team structure, total number of observations became 104.

TABLE 2 Demographics of survey respondents.

Title	Values	Number of records	Percentage
Population	DevOps engineer	74	(71.15%)
	Manager role (e.g., TEAM Lead, Manager, Senior Manager etc.)	12	(11.54%)
	Others (e.g., Developer, Operation Engineer, Architect, Project Manager, Consultant, and DevOps Trainers, etc.)	18	(17.31%)
Experience	0–2 Years	9	(8.65%)
	3–5 Years	23	(22.12%)
	6–10 Years	17	(16.35%)
	More than 10 Years	55	(52.88%)
Company size	1–9 Employees	5	(4.81%)
	10–49 Employees	15	(14.42%)
	50–249 Employees	24	(23.08%)
	More than 249 employees.	60	(57.69%)
Sectors	E-commerce	5	(4.81%)
	Telecommunication	8	(7.69%)
	Consulting and IT services	34	(32.69%)
	Financial/Fintech	37	(35.58%)
	Other (e.g., Insurance, Software, Automotive, and Retails, etc.).	20	(19.23%)
Company formations	Before 2000	48	(46.15%)
	Between 2000 and 2010	12	(11.54%)
	After 2010	44	(42.31%)
IT Unit/ department size	1–10 persons	21	(20.19%)
	11–20 persons	7	(6.73%)
	21–50 persons	18	(17.31%)
	51–100 persons	19	(18.27%)
	More than 100 persons	39	(37.50%)
DevOps team size	1–3 persons	28	(26.92%)
	4–7 persons	30	(28.85%)
	8–15 persons	25	(24.04%)
	16–30 persons	9	(8.65%)
	More than 30 persons	10	(9.62%)
	No members	2	(1.92%)

4 Statistical analyzing and the results

We used the IBM SPSS Statistics tool and generated the results based on appropriate statistical methods, such as Variance Analysis (ANOVA), t-test, and Chi-square test, based on the number of variables and conditions.

To compare DevOps team structures by their performance, 14 questions were being asked under the five performance-related metrics identified in the framework.

- Lead Time (LT)
- Deployment Frequency (DF)
- Mean Time To Repair/Recovery (MTTR)

2 https://docs.google.com/spreadsheets/d/16Cig8KReHwY6ZKWtICdNZDptIUY1we-n/edit?usp=drive_link&ouid=105145312824009983722&rtpof=true&sd=true

- Number of Incidents (NoI)
- Number of Failures/Service Interruptions (NoF/NoSI)

To be able to examine the temporal aspect of performance for RQ1 and RQ2, we not only asked the participants to provide values for these five software delivery performance metrics after DevOps adoption but also asked them to provide the same values before DevOps adoption. For RQ3, we also believe that due to some specific parameters such as the size and complexity of the company or application, high or low values given to these performance metrics may not reflect the actual performance of the teams. Thus, the participants are being asked to provide the percentage of their goal achievement for these performance criteria.

The relevant descriptive statistics for demographics are presented in Table 2.

a) The first performance-related metric concerns Lead Time (LT) when the request was made and ends when it was fulfilled. In this regard, the average LT "before and after moving DevOps" being asked to participants. We use a Paired Samples T test for RQ1 to test whether there is a significant difference between the average LT before and after DevOps adoption.

To understand whether implementing DevOps affects Lead Time, we develop two hypotheses. The null hypothesis (H0) assumes that the average Lead Time before and after adopting DevOps remains the same, meaning there is no effect. On the other hand, the alternative hypothesis (H1) suggests that the average Lead Time changes after DevOps is introduced, indicating a real impact. We then use statistical tests to see which hypothesis the data supports, helping us understand the influence of DevOps on team performance. The hypotheses for all four teams are as follows:

- H0: The mean values of the pre- and post-DevOps Lead Time are the same.
- H1: The mean values of the pre- and post-DevOps Lead Time are different.

The result in Table 3 shows that the significance values for TeamType2, TeamType3 and TeamType4 are less than the confidence level value (0.05); therefore, we reject the null hypothesis (H0) for these 3 teams, and conclude that there is a statistically significant difference for TeamType2, TeamType3 and TeamType4 Before and After DevOps for LT performance metric.

For TeamType1, the significance value of TeamType1 in Table 3 is 0.111, which is greater than the confidence level value of 0.05; therefore, we accept the null hypothesis. In this case, we say that there is no statistically significant difference between Before and After DevOps for TeamType1 from LT performance metric perspective.

For RQ2, when we can compare the mean values in Table 3, we see that TeamType3 has the greatest value which is −1.125 and we can state that TeamType3 is the best performer among the all four DevOps team formations from LT perspective after DevOps was adopted.

For RQ3, we will assess whether there are statistically significant differences in LT goal achievement percentage between the four DevOps groups. To compare the groups, we use variance analysis (ANOVA).

Descriptive statistics in Table 4 show LT goal means are close; therefore, we must statistically test if there is a meaningful difference.

ANOVA Hypothesis testing begins by testing if group variances are equal, using the Levene test. The hypotheses are:

- H0: The group variances are equal.
- H1: At least one group variance is different.

The significance value of 0.399 in Table 5 is greater than the significance level of 0.05. Thus, we accept the null hypothesis and state that the group variances are equal.

In this case, we ensured homogeneity assumptions; therefore, we use ANOVA to test group mean differentiation. The hypotheses are as follows:

- H0: Groups' LT goal achievement percentage means are equal.
- H1: At least one of the groups' LT goal achievement percentages is different.

The significance value of 0.370 in Table 6 is greater than the confidence level of 0.05. Therefore, we accept the null hypothesis and state that there is no statistical difference between the group Lead Time goal achievement percentage means. There is no significant performance difference between DevOps team formations from LT performance metric perspective.

b) The second performance-related test concerns Deployment Frequency (DF). We asked the participants to provide the average DF before and after moving DevOps. We use a Paired Samples T test for RQ1 to test whether there is a significant difference between the average DF before and after DevOps.

The hypotheses for all four teams are as follows:

- H0: Before and After DevOps DF means are equal.
- H1: Before and After DevOps DF means are not equal.

The result in Table 7 shows that the significance values for TeamType2, TeamType3 and TeamType4 are less than the confidence level value (0.05); therefore, we reject the null hypothesis (H0) for them and conclude that there is a statistically significant difference for TeamType2, TeamType3 and TeamType4 Before and After DevOps for DF performance metric.

For TeamType1, the significance value of TeamType1 in Table 7 is 0.154, which is greater than the confidence level value of 0.05; therefore, we accept the null hypothesis. In this case, we say that there is no statistically significant difference between Before and After DevOps for TeamType1 from DF performance metric perspective.

For RQ2, when we can compare the mean values in Table 7, we see that TeamType2 has the greatest value which is −1.444 and we can state that TeamType2 is the best performer among the all four DevOps team formations from DF perspective after DevOps was adopted.

10.3389/fcomp.2025.1554299

TABLE 3 Paired Sampled T tests for LT.

Paired Samples test										
How is your DevOps team organized?		Paired differences					t	df	Significance	
		Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				One-sided *p*	Two-sided *p*
					Lower	Upper				
TeamType4	What was your average Lead Time before you move DevOps (when the request is made and ends when it is fulfilled)?—What is your current average Lead Time after you move DevOps (when the request is made and ends when it is fulfilled)?	−1.107	1.133	0.214	−1.547	−0.668	−5.169	27	<0.001	<0.001
TeamType3	What was your average Lead Time before you move DevOps (when the request is made and ends when it is fulfilled)?—What is your current average Lead Time after you move DevOps (when the request is made and ends when it is fulfilled)?	−1.125	1.114	0.176	−1.481	−0.769	−6.389	39	<0.001	<0.001

(Continued)

10.3389/fcomp.2025.1554299

TABLE 3 (Continued)

Paired Samples test										
How is your DevOps team organized?		Paired differences					t	df	Significance	
		Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				One-sided *p*	Two-sided *p*
					Lower	Upper				
TeamType2	What was your average Lead Time before you move DevOps (when the request is made and ends when it is fulfilled)?—What is your current average Lead Time after you move DevOps (when the request is made and ends when it is fulfilled)?	−0.704	1.137	0.219	−1.154	−0.254	−3.215	26	0.002	0.003
TeamType1	What was your average Lead Time before you move DevOps (when the request is made and ends when it is fulfilled)?—What is your current average Lead Time after you move DevOps (when the request is made and ends when it is fulfilled)?	−0.778	1.302	0.434	−1.778	0.223	−1.793	8	0.055	0.111

TABLE 4 Descriptive statistics for lead time.

Descriptives								
What percentage did you achieve/reach the lead time goal?								
	N	Mean	Std. deviation	Std. error	95% confidence interval for mean		Minimum	Maximum
					Lower bound	Upper bound		
TeamType4	28	3.36	1.224	0.231	2.88	3.83	1	5
TeamType3	40	3.60	1.172	0.185	3.23	3.97	1	5
TeamType2	27	3.81	0.962	0.185	3.43	4.20	2	5
TeamType1	9	3.22	0.972	0.324	2.48	3.97	2	4
Total	104	3.56	1.122	0.110	3.34	3.78	1	5

TABLE 5 Tests of homogeneity of variances for LT.

Tests of homogeneity of variances					
		Levene statistic	df1	df2	Sig.
What percentage did you achieve/ reach the lead time goal?	Based on mean	0.994	3	100	0.399
	Based on median	0.852	3	100	0.469
	Based on median and with adjusted df	0.852	3	83.649	0.469
	Based on trimmed mean	0.980	3	100	0.405

TABLE 6 ANOVA for LT.

ANOVA					
What percentage did you achieve/reach the Lead Time goal?					
	Sum of squares	df	Mean square	F	Sig.
Between groups	3.996	3	1.332	1.060	0.370
Within groups	125.658	100	1.257		
Total	129.654	103			

To evaluate the goal achievement percentage between DevOps groups from a performance perspective for RQ3, we use variance analysis (ANOVA). This will determine whether there are statistically significant differences between the groups.

Table 8 shows that the DF goal means for each DevOps organization structure appear similar, but we must statistically test if the differences are significant.

As the first step in the ANOVA test, we again use the Levene test to test whether the group variances are equal. The hypotheses are as follows:

- H0: The group variances are equal.
- H1: At least one group variance is different.

The significance value of 0.005 in Table 9 is less than the confidence level value of 0.05; therefore, we reject the null hypothesis and say that the group variances are not equal. Because of the variance difference, the variance homogeneity assumption is rejected and the ANOVA test should not be used directly. We use the Brown-Welch test to determine whether group means are equal.

The hypotheses are as follows:

- H0: Groups DF goal achievement percentage means are equal.
- H1: At least one of the groups' DF goal achievement percentages is different.

The significance value of 0.004 in Table 10 is less than the confidence level of 0.05; therefore, we reject the null hypothesis and say that there is a statistical difference between the group DF goal achievement percentage means. DevOps organization structure types provide a significant difference in DF goal achievement. In this case, we need to examine the *Post hoc* test Tamhane in Table 11 to identify the difference.

Post hoc tests revealed that one group differed from the others. The hypothesis for TeamType2 is as follows:

- H0: DF goal achievement percentage mean for TeamType2 is equal to DF goal achievement percentage mean for TeamType4.
- H1: DF goal achievement percentage means are not equal between TeamType2 and TeamType4.

The significance value of 0.039 in Table 11 is less than the confidence level 0.05, indicating that the group means are not equal. There is a significant difference between TeamType2 and TeamType4, and TeamType2 (Mean is 4.19) performs better than TeamType4 (Mean is 3.39). However, when the same method is applied to other groups with a 0.005 confidence level, the *Post hoc* Tamhane test revealed no significant differences, and suggesting equal group means.

TABLE 7 Paired Sampled T tests for question DF.

Paired Samples test										
How your DevOps teams organized?		Paired differences					t	df	Significance	
		Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				One-sided *p*	Two-sided *p*
					Lower	Upper				
TeamType4	What was your deployment frequency before DevOps?—What is your current deployment frequency after DevOps?	−0.929	1.052	0.199	−1.336	−0.521	−4.673	27	<0.001	<0.001
TeamType3	What was your deployment frequency before DevOps?—What is your current deployment frequency after DevOps?	−1.425	1.130	0.179	−1.786	−1.064	−7.978	39	<0.001	<0.001
TeamType2	What was your deployment frequency before DevOps?—What is your current deployment frequency after DevOps?	−1.444	1.188	0.229	−1.914	−0.975	−6.320	26	<0.001	<0.001
TeamType1	What was your deployment frequency before DevOps?—What is your current deployment frequency after DevOps?	−0.778	1.481	0.494	−1.916	0.361	−1.575	8	0.077	0.154

TABLE 8 Descriptive statistics for DF.

Descriptives								
What percentage did you achieve/reach the deployment frequency goal?								
	N	Mean	Std. deviation	Std. error	95% confidence interval for mean		Minimum	Maximum
					Lower bound	Upper bound		
TeamType4	28	3.39	1.315	0.248	2.88	3.90	1	5
TeamType3	40	3.70	1.137	0.180	3.34	4.06	1	5
TeamType2	27	4.19	0.622	0.120	3.94	4.43	3	5
TeamType1	9	3.22	0.833	0.278	2.58	3.86	2	4
Total	104	3.70	1.096	0.107	3.49	3.92	1	5

TABLE 9 Tests of homogeneity of variances for DF.

Tests of homogeneity of variances					
		Levene statistic	df1	df2	Sig.
What percentage did you achieve/reach the deployment frequency goal?	Based on mean	4.603	3	100	0.005
	Based on median	4.116	3	100	0.008
	Based on median and with adjusted df	4.116	3	80.432	0.009
	Based on trimmed mean	4.191	3	100	0.008

TABLE 10 Brown-Welch test for DF.

Robust tests of equality of means				
What percentage did you achieve/reach the deployment frequency goal?				
	Statistic[a]	df1	df2	Sig.
Welch	5.426	3	32.728	0.004
Brown-Forsythe	3.709	3	68.942	0.016

[a]Asymptotically F distributed.

c) The next performance test measures Mean Time To Repair/Recovery (MTTR). We asked participants to provide MTTR before and after DevOps and use Paired Samples T test for RQ1 to determine if there is a significant difference between the two.

The hypotheses for all four teams are as follows:

- H0: Before and After DevOps MTTR means are equal.
- H1: Before and After DevOps MTTR means are not equal.

The result in Table 12 shows that the significance values for all four DevOps team structures are less than the confidence level value (0.05); therefore, we reject the null hypothesis (H0) for all and conclude that there is a statistically significant difference Before and After DevOps for MTTR performance metric.

For RQ2, when we can compare the mean values in Table 12, we see that TeamType4 has the greatest value which is −1.036 and we can state that TeamType4 is the best performer among the all four DevOps team formations from MTTR perspective after DevOps was adopted.

For RQ3, we will assess whether there are significant differences in the application repair/recovery SLA (MTTR) goal achievement percentage among the four DevOps groups. To compare the groups, we use variance analysis (ANOVA).

The MTTR goal means in Table 13 for each DevOps organization structure are similar, but we need to determine whether the differences are statistically significant.

Once again, we use Levene's test to determine whether the group variances are equal as the first step in the ANOVA Hypothesis testing. The hypotheses are as follows:

- H0: The group variances are equal.
- H1: At least one group variance is different.

The significance value of 0.017 in Table 14 is less than the confidence value of 0.005. Consequently, we reject the null hypothesis and state that group variances are not equal.

As the variance homogeneity assumption is rejected, the ANOVA test should not be used directly. Instead, we use the Brown-Welch Test to determine whether the group means are equal.

The hypotheses are as follows:

- H0: Group application repair/recovery SLA (MTTR) goal means are equal.
- H1: At least one of the group application repair/recovery SLA (MTTR) goal means is different.

The significance value of 0.878 in Table 15 is greater than the confidence level of 0.05. Therefore, we accept the null hypothesis and state that there is no statistical difference between group

TABLE 11 The *post hoc* test Tamhane for DF.

Multiple comparisons							
Dependent variable: What percentage did you achieve/reach the deployment frequency goal?							
	(I) How your DevOps teams organized?	(J) How your DevOps teams organized?	Mean difference (I-J)	Std. error	Sig.	95% confidence interval	
						Lower bound	Upper bound
Tamhane	TeamType4	TeamType3	−0.307	0.307	0.902	−1.15	0.53
		TeamType2	−0.792*	0.276	0.039	−1.56	−0.03
		TeamType1	0.171	0.373	0.998	−0.91	1.25
	TeamType3	TeamType4	0.307	0.307	0.902	−0.53	1.15
		TeamType2	−0.485	0.216	0.158	−1.07	0.10
		TeamType1	0.478	0.331	0.670	−0.52	1.47
	TeamType2	TeamType4	0.792*	0.276	0.039	0.03	1.56
		TeamType3	0.485	0.216	0.158	−0.10	1.07
		TeamType1	0.963	0.303	0.050	0.00	1.93
	TeamType1	TeamType4	−0.171	0.373	0.998	−1.25	0.91
		TeamType3	−0.478	0.331	0.670	−1.47	0.52
		TeamType2	−0.963	0.303	0.050	−1.93	0.00

* The mean difference is significant at the 0.05 level.

application repair/recovery SLA (MTTR) goal means. There is no significant performance difference between DevOps team formations from MTTR performance metric perspective. DevOps organization structure types do not provide a significant difference in application repair/recovery SLA (MTTR) goal achievement.

d) We test the Number of Incidents (NoI) using a Paired Samples T test for RQ1 to determine if there is a significant difference in the average DevOps incidents before and after.

The hypotheses for all four teams are as follows:

- H0: Before and After DevOps NoI means are equal.
- H1: Before and After DevOps NoI means are not equal.

The result in Table 16 shows that the significance values for TeamType2, TeamType3 and TeamType4 are less than the confidence level value (0.05); therefore, we reject the null hypothesis (H0) for TeamType2, TeamType3 and TeamType4, and conclude that there is a statistically significant difference Before and After DevOps for NoI performance metric.

For TeamType1, the significance value of TeamType1 in Table 16 is 0.860, which is greater than the confidence level value of 0.05; therefore, we accept the null hypothesis. In this case, we say that there is no statistically significant difference between Before and After DevOps for TeamType1 from NoI performance metric perspective.

For RQ2, when we can compare the mean values in Table 16, we see that TeamType4 has the greatest value which is −1.185 and we can state that TeamType4 is the best performer among the all four DevOps team formations from MTTR perspective after DevOps was adopted.

e) To test the Number of Failures/Service Interruptions (NoF/NoSI), we use the Paired Samples T test for RQ1 to determine if there is a significant difference in the average NoF/NoSI before and after DevOps.

The hypotheses for all four teams are as follows:

- H0: Before and After DevOps NoF/NoSI means are equal.
- H1: Before and After DevOps NoF/NoSI means are not equal.

The result in Table 17 shows that the significance values for TeamType2, TeamType3 and TeamType4 are less than the confidence level value (0.05); therefore, we reject the null hypothesis (H0) for them and conclude that there is a statistically significant difference for TeamType2, TeamType3 and TeamType4 Before and After DevOps from NoF/NoSI performance metric perspective.

For TeamType1, the significance value of TeamType1 in Table 17 is 0.559, which is greater than the confidence level value of 0.05; therefore, we accept the null hypothesis. In this case, we say that there is no statistically significant difference between Before and After DevOps for TeamType1 from NoF/NoSI performance metric perspective.

For RQ2, when we can compare the mean values in Table 17, we see that TeamType2 has the greatest value which is −1.111 and we can state that TeamType2 is the best performer among the all four DevOps team formations from NoF/NoSI perspective after DevOps was adopted.

For RQ3, we will assess whether there are statistically significant differences in (NoF/NoSI) the application availability Service Level Agreement (SLA) goal achievement percentage between four DevOps groups. To compare groups, we use variance analysis (ANOVA).

10.3389/fcomp.2025.1554299

TABLE 12 Paired Sampled T tests for MTTR.

How your DevOps teams organized?		Paired Samples test								
		Paired differences					t	df	Significance	
		Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				One-sided *p*	Two-sided *p*
					Lower	Upper				
TeamType4	What was the average mean time for repair/recovery before DevOps?—What is the average current mean time for repair/recovery after DevOps?	−1.036	0.881	0.167	−1.377	−0.694	−6.220	27	<0.001	<0.001
TeamType3	What was the average mean time for repair/recovery before DevOps?—What is the average current mean time for repair/recovery after DevOps?	−0.850	0.736	0.116	−1.085	−0.615	−7.309	39	<0.001	<0.001
TeamType2	What was the average mean time for repair/recovery before DevOps?—What is the average current mean time for repair/recovery after DevOps?	−1.000	1.144	0.220	−1.452	−0.548	−4.544	26	<0.001	<0.001
TeamType1	What was the average mean time for repair/recovery before DevOps?—What is the average current mean time for repair/recovery after DevOps?	−1.000	1.118	0.373	−1.859	−0.141	−2.683	8	0.014	0.028

TABLE 13 Descriptive statistics for MTTR.

Descriptives								
What percentage did you meet the SLA for application repair/recovery?								
	N	Mean	Std. deviation	Std. error	95% confidence interval for mean		Minimum	Maximum
					Lower bound	Upper bound		
TeamType4	28	4.07	1.120	0.212	3.64	4.51	1	5
TeamType3	40	4.08	1.071	0.169	3.73	4.42	1	5
TeamType2	27	4.04	1.091	0.210	3.61	4.47	2	5
TeamType1	9	3.56	1.810	0.603	2.16	4.95	1	5
Total	104	4.02	1.157	0.113	3.79	4.24	1	5

TABLE 14 Tests of homogeneity of variances for MTTR.

Tests of homogeneity of variances					
		Levene statistic	df1	df2	Sig.
What percentage did you meet the SLA for application repair/recovery?	Based on mean	3.580	3	100	0.017
	Based on median	1.546	3	100	0.207
	Based on median and with adjusted df	1.546	3	47.641	0.215
	Based on trimmed mean	3.436	3	100	0.020

The application availability SLA goal from NoF/NoSI perspective in Table 18 for each DevOps organizational structure is similar, but we need to test whether the difference is statistically significant.

As the first step of ANOVA Hypothesis, we use Levene's test to determine whether the group variances are equal. The hypotheses are as follows:

- H0: The group variances are equal.
- H1: At least one group variance is different.

The significance value of 0.660 in Table 19 is greater than the confidence value of 0.05; therefore, we accept the null hypothesis and state that the group variances are equal.

As we ensured the homogeneity assumptions, we use ANOVA directly and test the groups' means of differentiation.

The hypotheses are as follows:

- H0: Groups (NoF/NoSI) the application availability SLA goal achievement percentage means are equal.
- H1: At least one of the groups (NoF/NoSI) the application availability SLA goal achievement percentage mean is different.

TABLE 15 Brown-Welch test for MTTR.

Robust tests of equality of means				
What percentage did you meet the SLA for application repair/recovery?				
	Statistic[a]	df1	df2	Sig.
Welch	0.225	3	30.238	0.878
Brown-Forsythe	0.390	3	25.315	0.761

[a]Asymptotically F distributed.

The significance value of 0.761 in Table 20 is less than the confidence level of 0.05, so we accept the null hypothesis and say that there is no statistical difference between groups in (NoF/NoSI) the application availability SLA goal achievement percentage. DevOps organization structure types do not provide a significant difference for (NoF/NoSI) the application availability SLA goal achievement.

5 Limitations

Our study is limited to comparing four different types of DevOps formations based on five different performance metrics. We also tested whether DevOps adoption makes any differences by examining the same performance metrics values before and after DevOps adoption. Due to the scope of our survey questions, we did not go deep into the other rationales behind our topic such as why there are different types of DevOps team formations and why companies are selecting any of them. We believe that these questions call for new promising research topics.

6 Discussion

In this study, we examine DevOps team structures and determine their performance differences and propose the framework of Formation-Performance of DevOps teams. To measure the difference between Pre-DevOps and Post-DevOps and find the best performer among the four DevOps team formations, we identified the following five performance metrics: "Lead Time (LT), Deployment Frequency (DF), Mean Time to Recovery (MTTR),

10.3389/fcomp.2025.1554299

TABLE 16 Paired Sampled T tests for NoI.

Paired Samples test										
How your DevOps teams organized?		Paired differences					t	df	Significance	
		Mean	Std. deviation	Std. error mean	95% Confidence interval of the difference				One-sided *p*	Two-sided *p*
					Lower	Upper				
TeamType4	What was the average number of incidents in a month before DevOps?—What is the current average number of incidents in a month after DevOps?	−0.714	1.301	0.246	−1.219	−0.210	−2.905	27	0.004	0.007
TeamType3	What was the average number of incidents in a month before DevOps?—What is the current average number of incidents in a month after DevOps?	−0.900	1.236	0.195	−1.295	−0.505	−4.604	39	<0.001	<0.001
TeamType2	What was the average number of incidents in a month before DevOps?—What is the current average number of incidents in a month after DevOps?	−1.185	1.039	0.200	−1.596	−0.774	−5.927	26	<0.001	<0.001
TeamType1	What was the average number of incidents in a month before DevOps?—What is the current average number of incidents in a month after DevOps?	−0.111	1.833	0.611	−1.520	1.298	−0.182	8	0.430	0.860

10.3389/fcomp.2025.1554299

TABLE 17 Paired Sampled T tests for NoF/NoSI.

Paired Samples test										
How your DevOps teams organized?		Paired differences					t	df	Significance	
		Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				One-sided *p*	Two-sided *p*
					Lower	Upper				
TeamType4	What was the number average of failures/service interruption in a month before DevOps?—What is the current average number of failures/service interruption in a month after DevOps?	−0.679	1.020	0.193	−1.074	−0.283	−3.519	27	<0.001	0.002
TeamType3	What was the number average of failures/service interruption in a month before DevOps?—What is the current average number of failures/service interruption in a month after DevOps?	−0.775	0.832	0.131	−1.041	−0.509	−5.894	39	<0.001	<0.001
TeamType2	What was the number average of failures/service interruption in a month before DevOps?—What is the current average number of failures/service interruption in a month after DevOps?	−1.111	1.050	0.202	−1.526	−0.696	−5.498	26	<0.001	<0.001
TeamType1	What was the number average of failures/service interruption in a month before DevOps?—What is the current average number of failures/service interruption in a month after DevOps?	−0.222	1.093	0.364	−1.062	0.618	−0.610	8	0.279	0.559

TABLE 18 Descriptive statistics for NoF/NoSI.

Descriptives								
What percentage did you meet the SLA for the application availability?								
	N	Mean	Std. deviation	Std. Error	95% confidence interval for mean		Minimum	Maximum
					Lower bound	Upper bound		
TeamType4	28	4.39	0.994	0.188	4.01	4.78	1	5
TeamType3	40	4.13	1.181	0.187	3.75	4.50	1	5
TeamType2	27	4.30	1.103	0.212	3.86	4.73	1	5
TeamType1	9	4.11	1.054	0.351	3.30	4.92	3	5
Total	104	4.24	1.093	0.107	4.03	4.45	1	5

TABLE 19 Tests of homogeneity of variances for NoF/NoSI.

Tests of homogeneity of variances					
		Levene statistic	df1	df2	Sig.
What percentage did you meet the SLA for the application availability?	Based on mean	0.534	3	100	0.660
	Based on median	0.389	3	100	0.761
	Based on median and with adjusted df	0.389	3	98.187	0.761
	Based on trimmed mean	0.619	3	100	0.604

TABLE 20 Tests of homogeneity of variances for NoF/NoSI.

ANOVA					
What percentage did you meet the SLA for the application availability?					
	Sum of squares	df	Mean square	F	Sig.
Between groups	1.418	3	0.473	0.389	0.761
Within groups	121.572	100	1.216		
Total	122.990	103			

Number of Incidents (NoI) and Number of Failures/Service Interruptions (NoF/NoSI)." The four different DevOps team formations that we identified are follows: "Separate Development and Operation teams with limited collaboration (TeamType1)," "Separate Development and Operation teams with high collaboration (TeamType2)," "Separate Development and Operation teams that are supported by a DevOps/Platform/Infra team (TeamType3)," and "A single team that includes both Development and Operations (TeamType4)." We intentionally avoided naming DevOps team structure types because naming conventions can be variable and named differently by company. Instead, we define DevOps team structures based on the level of collaboration and organizational relationships.

A quantitative survey was conducted to evaluate performance changes of four distinct DevOps team types before and after DevOps adoption across five key metrics. Initial recruitment through snowball sampling (Biernacki and Waldorf, 1981) and open-call crowdsourcing (Kosinski et al., 2015) yielded limited responses. To improve participation, targeted advertising (Thornton et al., 2016) through LinkedIn outreach was employed. While identifying potential participants, we did not apply any specific filters such as experience level, industry sector, country, gender, or age; the only selection criterion was that the profile indicated expertise or involvement in DevOps. This approach resulted in 105 completed responses from 380 outreach attempts, achieving a 32.1% initial response rate and a 69.7% completion rate.

In Section 4, we reported whether the differences in performance metrics before and after DevOps adoption were statistically significant. In this section, we further assess the significance of those findings with the 95% Confidence Intervals (CI) by presenting the actual effect sizes based on the differences between pre- and post-DevOps mean values as shown in Table 21.

In the paired samples tests for each performance metric, the results include the lower and upper bounds of the 95% CI for the

TABLE 21 Actual effect.

Metric	Team	Pre-mean	Post-mean	Mean diff.	%95 confidence Interval of the difference	
					Lower	Upper
LT	TeamType1	3.33	4.11	−0.78	−1.78	0.22
	TeamType2	3.44	4.15	−0.71	−1.15	−0.25
	TeamType3	3.15	4.28	−1.13	−1.48	−0.77
	TeamType4	3.46	4.57	−1.11	−1.55	−0.67
DF	TeamType1	2.33	3.11	−0.78	−1.92	0.36
	TeamType2	2.26	3.70	−1.44	−1.91	−0.98
	TeamType3	2.38	3.80	−1.42	−1.79	−1.06
	TeamType4	2.93	3.86	−0.93	−1.34	−0.52
MTTR	TeamType1	3.00	4.00	−1.00	−1.86	−0.14
	TeamType2	3.04	4.04	−1.00	−1.45	−0.55
	TeamType3	2.78	3.62	−0.84	−1.09	−0.62
	TeamType4	2.86	3.89	−1.03	−1.38	−0.69
NoI	TeamType1	4.00	4.11	−0.11	−1.52	1.30
	TeamType2	3.26	4.44	−1.18	−1.60	−0.77
	TeamType3	3.62	4.53	−0.91	−1.30	−0.51
	TeamType4	3.64	4.36	−0.72	−1.22	−0.21
NoF-NoSI	TeamType1	4.44	4.67	−0.23	−1.06	0.62
	TeamType2	3.85	4.96	−1.11	−1.53	−0.70
	TeamType3	3.98	4.75	−0.77	−1.04	−0.51
	TeamType4	3.89	4.57	−0.68	−1.07	−0.28

TABLE 22 Standardized effect size.

Metric	Team	Pre-mean	Pre-SD	Post-mean	Post-SD	Mean diff.	Pooled-SD	Cohens-d	Effect
LT	TeamType1	3.33	1.50	4.11	1.36	−0.78	1.432	−0.54	Medium
	TeamType2	3.44	1.31	4.15	1.03	−0.71	1.178	−0.60	Medium
	TeamType3	3.15	1.31	4.28	0.78	−1.13	1.078	−1.05	Large
	TeamType4	3.46	1.35	4.57	0.79	−1.11	1.106	−1.00	Large
DF	TeamType1	2.33	1.41	3.11	1.62	−0.78	1.519	−0.51	Medium
	TeamType2	2.26	1.20	3.70	1.56	−1.44	1.392	−1.03	Large
	TeamType3	2.38	1.23	3.80	1.14	−1.42	1.186	−1.20	Large
	TeamType4	2.93	1.39	3.86	1.30	−0.93	1.346	−0.69	Medium
MTTR	TeamType1	3.00	1.22	4.00	1.00	−1.00	1.115	−0.90	Large
	TeamType2	3.04	1.09	4.04	1.06	−1.00	1.075	−0.93	Large
	TeamType3	2.78	0.95	3.62	1.03	−0.84	0.991	−0.85	Large
	TeamType4	2.86	0.85	3.89	0.83	−1.03	0.840	−1.23	Large
NoI	TeamType1	4.00	1.32	4.11	1.45	−0.11	1.387	−0.08	Negligible
	TeamType2	3.26	1.10	4.44	0.89	−1.18	1.001	−1.18	Large
	TeamType3	3.62	1.21	4.53	0.85	−0.91	1.046	−0.87	Large
	TeamType4	3.64	1.37	4.36	0.83	−0.72	1.133	−0.64	Medium
NoF-NoSI	TeamType1	4.44	0.73	4.67	1.00	−0.23	0.875	−0.26	Small
	TeamType2	3.85	1.06	4.96	0.19	−1.11	0.761	−1.46	Large
	TeamType3	3.98	1.00	4.75	0.59	−0.77	0.821	−0.94	Large
	TeamType4	3.89	1.13	4.57	0.63	−0.68	0.915	−0.74	Medium

mean difference. If the CI does not include zero, it indicates statistical significance—providing evidence that the observed difference is unlikely to have occurred by chance. Furthermore, if the entire interval lies below zero, it suggests a statistically significant improvement in that performance metric. Upon examining these intervals in Table 21, we find strong evidence of real performance improvements for all DevOps team formations except TeamType1. For TeamType1, only the MTTR metric shows a confidence interval that excludes zero, suggesting a statistically significant positive effect; the remaining metrics include zero in their intervals, indicating no significant difference. These findings are consistent with the statistical results discussed in Section 4.

To better understand the magnitude of these effects, we also calculated Cohen's *d* values (Cohen, 1988), which represent standardized effect sizes. These values are obtained by dividing the mean difference by the standard deviation and are summarized in Table 22. Cohen's *d* provides an estimate of the significance of the observed differences, complementing the *p*-values reported earlier. Unlike p-values, which merely indicate whether an effect exists, Cohen's *d* quantifies how substantial that effect is. Its scale-independent nature allows for meaningful comparisons across different metrics and studies, and it is widely used in meta-analyses to synthesize empirical results. The interpretation of Cohen's *d* values follows commonly accepted benchmarks (presented in Table 23).

Based on the values in Table 22 and the classification above, the standardized effect sizes for each team type and performance metric are summarized as follows:

- Lead Time (LT): TeamType3 and TeamType4 exhibit "Large" effects, while TeamType1 and TeamType2 show "Medium" effects.
- Deployment Frequency (DF): TeamType2, TeamType3 show "Large" effects; TeamType1 and TeamType4 have "Medium" effect.
- Mean Time to Recovery (MTTR): All team types demonstrate "Large" effects.
- Number of Incidents (NoI): TeamType2 and TeamType3 shows a "Large" effect; TeamType4 exhibits "Medium" effects; TeamType1's effect is "Negligible."
- Number of Failures/Service Interruptions (NoF/NoSI): TeamType2 and TeamType3 show "Large" effects, TeamType4 a "Medium" effect, and TeamType1 a "Small" effect.

These results highlight that, while most team formations benefited significantly from DevOps adoption, the extent of improvement varies across metrics and team types, with TeamType1 generally showing weaker or negligible improvements. Figure 2 presents violin plots comparing pre- and post-DevOps values for five key performance metrics—Lead Time (LT), Deployment Frequency (DF), Mean Time to Recovery (MTTR), Number of Incidents (NoI), and Number of Failures/Service Interruption (NoF-NoSI)—across DevOps team types. Each plot illustrates the

TABLE 23 Cohen's effect size guidelines.

Cohen's d	Effect size
0.01–0.19	Negligible
0.20–0.49	Small
0.50–0.79	Medium
0.8 and above	Large

TABLE 24 DevOps adoption team performance comparison.

	Lead-time (LT)	Deployment frequency (DF)	Mean time to recovery (MTTR)	Number of incidents (NoI)	Number of failures/ service interruptions (NoF/NoSI)
TeamType1—Separate development and operation teams with limited collaboration	No	No	Yes	No	No
TeamType2—Separate development and operation teams with high collaboration	Yes	Yes—Better	Yes	Yes—Better	Yes—Better
TeamType3—Separate Development and Operation teams that are supported by a DevOps/ platform/infra team	Yes—Better	Yes	Yes	Yes	Yes
TeamType4—A single team that includes both development and operations	Yes	Yes	Yes—Better	Yes	Yes

* "Yes" means that the team shows performance improvement comparing to Pre and Post DevOps values. ** "Better" means that the team shows bettter performance for that metric comparing to the others.

TABLE 25 DevOps team performance comparison.

	Lead-time (LT)	Deployment frequency (DF)	Mean time to recovery (MTTR)	Number of failures/ service interruptions (NoF/NoSI)
TeamType1—Separate development and operation teams with limited collaboration	No	No	No	No
TeamType2—Separate development and operation teams with high collaboration	No	Yes (between TeamType4) Better	No	No
TeamType3—Separate development and operation teams that are supported by a DevOps/platform/infra team	No	No	No	No
TeamType4—A single team that includes both development and operations	No	Yes (between TeamType2)	No	No

* "Yes" means that there is a performance difference between the given teams. It also indicates which team shows better performance than the others from performance perspective.

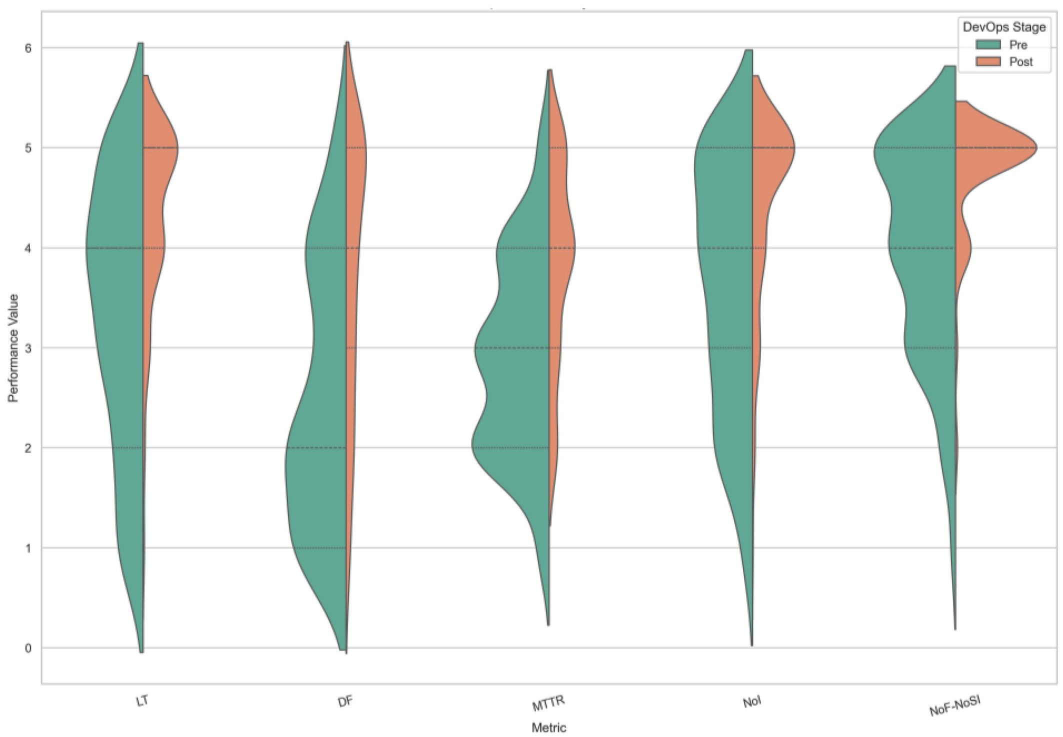

FIGURE 2
DevOps performance metrics: violin plots and effect size analysis based on Cohen's d values.

distribution and central tendencies before and after DevOps adoption.

Based on our test results, we found that there is a statistically significant difference between Before and After DevOps for all teams except TeamType1 for the LT, DF, NoI, and NoF/NoSI performance metrics, and they perform better after moving to DevOps. This suggests that adopting the DevOps approach increases the performance of the team and efficiency for all types of DevOps team formations except TeamType1. This conclusion also addresses our first research question (RQ1).

We also identify that while TeamType3 is slightly better than the others from the LT perspective, TeamType2 is slightly better than the others from the DF, NoI, and NoF/NoSI perspectives. For MTTR test, we found that there is a statistically significant difference between Before and After for all DevOps groups and perform better after moving to DevOps. TeamType4 is the best performer for this metric. The Pre-DevOps and Post-DevOps adoption performance results of the four teams are summarized in Table III. From our second research question perspective (RQ2), we found that TeamType2 performs better for 3 performance metrics out of 5 as shown in Table 24. Hence, we can conclude that TeamType2 is the best performer among the four DevOps Team formations in terms of DevOps adoption.

To address our last research question (RQ3), we found that DevOps organization structure types do not provide a significant difference for LT, NoF/NoSI, and MTTR from the goal achievement perspective. However, there is a statistical difference between TeamType2 and TeamType4 from the DF metric and TeamType2 is better than TeamType4. The performance comparisons of DevOps teams are summarized in Table 25.

7 Conclusion

In conclusion, we show that DevOps adoption makes a significant difference no matter which DevOps formation type is realized except TeamType1. Regarding TeamType1, DevOps adoption only makes a difference in the Mean Time To Repair/Recovery (MTTR) performance metric. While TeamType3's Lead Time (LT) performance and TeamType4's MTTR performance are better than the others, TeamType2's Deployment Frequency (DF), Number of Incident (NoI) and Number of Failures/Service Interruptions (NoF/NoSI) performance metrics are better than the other teams. However, after DevOps adoption, there is no performance difference in LT, MTTR and NoF/NoSI performance metrics among these four DevOps team types. From the DF perspective, there is only a performance difference between TeamType2 and TeamType4, and TeamType2 is better than TeamType4.

In the present study, we focus only on the performance differences of the four types of DevOps team structures. One needs to articulate the reasons behind the different type of DevOps team formations and selection which can bring out new insights for future study for researchers and practitioners.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article: https://docs.google.com/spreadsheets/d/16Cig8KReHwY6ZKWtICdNZDptIUY1we-n/edit?usp=drive_link&ouid=105145312824009983722&rtpof=true&sd=true.

Ethics statement

Written informed consent was not obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article because the survey was completed anonymously, and we did not ask for any personal or company-related information from participants except for an email address that was not mandatory and asked only if they wanted to obtain the survey results.

Author contributions

HK: Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. MA: Supervision, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that Gen AI was used in the creation of this manuscript. Used to improve readability.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. (2016). *Site reliability engineering: how Google runs production systems*. Sebastopol, CA: O'Reilly Media.

Biernacki, P., and Waldorf, D. (1981). Snowball sampling—problems and techniques of chain referral sampling. *Sociol. Methods Res.* 10, 141–163. doi: 10.1177/004912418101000205

Brickman Bhutta, C. (2012). Not by the book: Facebook as a sampling frame. *Sociol. Methods Res.* 41, 57–88. doi: 10.1177/0049124112440795

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. *2nd* Edn. London: Routledge.

Debois, P. (2008) Agile infrastructure and operations: How infra-gile are you? In Agile 2008 conference, pp. 202–207.

Díaz, J., López-Fernández, D., Pérez, J., and González-Prieto, Á. (2021). Why are many businesses instilling a DevOps culture into their organization? *Empir. Softw. Eng.* 26:25. doi: 10.1007/s10664-020-09919-3

Díaz, J., Pérez, J., Alves, I., Kon, F., Leite, L., Meirelles, P., et al. (2024). Harmonizing devOps taxonomies — a grounded theory study. *J. Syst. Softw.* 208:111908. doi: 10.1016/j.jss.2023.111908

DORA DORA's software delivery metrics: the four keys. Available online at: https://dora.dev/guides/dora-metrics-four-keys/ (Accessed September 01, 2022).

Faul, F., Erdfelder, E., Lang, A. G., and Buchner, A. (2007). G*power 3: a flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behav. Res. Methods* 39, 175–191. doi: 10.3758/BF03193146

Glaser, B., and Strauss, A. (1999). *The discovery of grounded theory: Strategies for qualitative research*. New Jersey: Aldine Transaction.

Gomes, M., Pereira, R., Silva, M., Braga de Vsconcelos, J., and Rocha, Á. (2022). *KPI's for Evaluation of DevOps Teams*. doi: 10.1007/978-3-031-04829-6_13

Gray, J. (2006). A conversation with Werner Vogels. *ACM Queue* 4, 14–22.

Heckathorn, D. D. (1997). Respondent-driven sampling: a new approach to the study of hidden populations. *Soc. Probl.* 44, 174–199. doi: 10.2307/3096941

Hüttermann, M. (2012). *DevOps for Developers*. Berkeley, CA: Apress.

Jayakody, V., and Wijayanayake, J. (2023). Critical success factors for DevOps adoption in information systems development. *Int. J. Inf. Syst. Proj. Manag.* 11:Article 5.

Kim, G., Humble, J., Debois, P., and Willis, J. *The DevOps handbook: how to create world-class agility, reliability, and security in technology organizations*. Portland, Oregon: IT Revolution (2016)

Kniberg, H. Spotify engineering culture (part 1), (2014) Available online at: https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1 (Accessed September 01, 2023).

Kosinski, M., Matz, S. C., Gosling, S. D., Popov, V., and Stillwell, D. (2015). Facebook as a research tool for the social sciences: opportunities, challenges, ethical considerations, and practical guidelines. *Am. Psychol.* 70, 543–556. doi: 10.1037/a0039210

Leite, L., Kon, F., Pinto, G., and Meirelles, P. "Building a theory of software teams Organization in a Continuous Delivery Context," IEEE/ACM 42nd international conference on software engineering: Companion proceedings (ICSE-companion), Seoul, Korea (south), (2020), pp. 296–297.

Leite, L., Lago, N., Melo, C., Kon, C., and Meirelles, P. (2023). A theory of organizational structures for development and infrastructure professionals. *IEEE Trans. Softw. Eng.* 49, 1898–1911. doi: 10.1109/TSE.2022.3199169

Leite, L., Pinto, G., Kon, F., and Meirelles, P. (2021). The organization of software teams in the quest for continuous delivery: a grounded theory approach. *Inf. Softw. Technol.* 139:106672. doi: 10.1016/j.infsof.2021.106672

Leite, L., Rocha, C., Kon, F., Milojicic, D., and Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Comput. Surv.* 52:127. doi: 10.1145/3359981

López-Fernández, D., Díaz, J., García, J., Pérez, J., and González-Prieto, Á. (2022). "DevOps Team Structures: Characterization and Implications," in *IEEE Transactions on Software Engineering*. 48, 3716–3736. doi: 10.1109/TSE.2021.3102982

Loukides, M. (2012). *What is DevOps?* Sebastopol, CA: O'Reilly Media.

Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2016a). *An exploratory study of DevOps: extending the dimensions of DevOps with practices*. Rome, Italy: The Eleventh International Conference on Software Engineering Advances (ICSEA).

Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2016b). "Relationship of DevOps to agile, lean and continuous deployment," in *Product-focused software process improvement. PROFES 2016*. eds. P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, and T. Mikkonen (Springer, Cham: Lecture Notes in Computer Science), 10027. doi: 10.1007/978-3-319-49094-6_27

Muñoz, M., Mejia, J., and Laporte, C. Y. (2019). "Reinforcing Very Small Entities Using Agile Methodologies with the ISO/IEC 29110" in *Trends and applications in software engineering, CIMPS 2018. Advances in intelligent systems and computing*. eds. J. Mejia, M. Muñoz, Á. Rocha, A. Peña and M. Pérez-Cisneros (Cham: Springer).

Muñoz, M., Negrete, M., and Arcilla-Cobián, M. (2021). Using a platform based on the basic profile of ISO/IEC 29110 to reinforce DevOps environments. *JUCS* 27, 91–110. doi: 10.3897/jucs.65080

Plant, O., Hillegersberg, J., and Aldea, A. (2022). Design and validation of a capability measurement instrument for DevOps teams. Agile Processes in Software Engineering and Extreme Programming, 23rd International Conference on Agile Software Development.

Rossi, S. (2012). "Model checking adaptive multilevel service compositions" in *Formal aspects of component software. FACS 2010*. eds. L. S. Barbosa and M. Lumpe (Berlin, Heidelberg: Springer).

Senapathi, M., Buchan, J., and Osman, H. (2018). "DevOps capabilities, practices, and challenges: Insights from a case study," in *Proceedings of the 22nd international conference on Evaluation and Assessment in Software Engineering 2018 (EASE '18)*. New York, NY, USA: Association for Computing Machinery, 57–67. doi: 10.1145/3210459.3210465

Shahin, M., Zahedi, M., Ali Babar, M., and Zhu, L. (2017). "Adopting continuous delivery and deployment: impacts on team structures, collaboration and responsibilities" in *Proceedings of 21st international conference on evaluation and assessment in software engineering, Karlskrona Sweden, (June 15 16, 2017 EASE 17)*.

Thornton, L., Batterham, P. J., Fassnacht, D. B., Kay-Lambkin, F., Calear, A. L., and Hunt, S. (2016). Recruiting for health, medical or psychosocial research using Facebook: systematic review. *Internet Interv.* 4, 72–81. doi: 10.1016/j.invent.2016.02.001

Williams, F., Nancy, J., Andrewson, S., and Richard, H. (2025). *Measuring DevOps success: Key metrics and KPIs*. Available at: https://www.researchgate.net/publication/391240883_Measuring_DevOps_Success_Key_Metrics_and_KPIs

Zhou, X., Huang, H., Zhang, H., Huang, X., and Zhong, C. (2022). *A cross-company ethnographic study on software teams for DevOps and microservices: organization, benefits, and issues*. doi: 10.1145/3510457.3513054