Information and Software Technology 177 (2025) 107569

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits

Fatiha El Aouni [a], Karima Moumane [a,*], Ali Idri [a,b], Mehdi Najib [c], Saeed Ullah Jan [d]

[a] Software Project Management Research Team, ENSIAS, Mohammed V University in Rabat, 10112 Rabat, Morocco
[b] MSDA, University Mohamed VI Polytechnic, Ben GuerirMorocco, Morocco
[c] TICLab, College of Engineering and Architecture, International University of Rabat, Sala Al Jadida, Morocco
[d] Higher Education Department, at Govt Degree College Wari Dir Upper, 18200, Pakistan

ARTICLE INFO

Keywords:
DevOps
Cloud
Agile
Integration
Synergy

ABSTRACT

Context: In today's fast-paced digital landscape, integrating DevOps, cloud, and agile methodologies is crucial for meeting software demands. However, this integration remains under-researched.
Objective: This study explores the integration of Agile, Cloud, and DevOps in today's software development landscape. It aims to analyze the challenges and benefits associated with merging these three approaches, focusing on their impact on software testing and the role of mindset in successful implementation and identifying the most suitable Agile methodologies.
Methods: This investigation utilizes a Systematic Literature Review(SLR) to enrich comprehension of this integration in current software development practices.
Results: The analysis of 31 articles highlights benefits such as improved collaboration and accelerated development, despite challenges with tool proliferation. Platforms like Jenkins, GitLab, Kubernetes, and Docker show promise in addressing these complexities. Our study examines the advantages and challenges of this integration, focusing on its impact on software testing and the role of mindset in successful implementation and identifying the most suitable Agile methodologies.
Conclusion: The integration of Agile, DevOps, and Cloud signifies a vital move towards collaborative, scalable, and automated methods, crucial for swift delivery, enhanced quality, and ongoing competitiveness. This unified approach is fundamental for organizational advancement and innovation in the ever-evolving software development realm. Further research should tackle challenges in merging these methods and delve into their interactions with emerging technologies to refine practices for increased efficiency.

1. Introduction

In today's rapidly changing digital world, companies encounter significant challenges in making and delivering software that meets their customers' ever-shifting demands [1,2]. Successful companies adopt a holistic approach combining DevOps, cloud computing, and agile methodologies to tackle these challenges. By putting these three solid and teamwork-focused strategies together [3], companies try to make their software development and delivery better, help teams work together more effectively, and stay flexible to changing market needs. This way of working helps companies remain competitive, innovate, and provide better customer service in our fast-changing digital world [4].

Combining Agile, Cloud, and DevOps methodologies offers a powerful and synergistic approach to modern software development and IT operations. The integration of these practices brings forth numerous benefits that contribute to enhanced efficiency, faster development cycles, and improved overall productivity for organizations [5–7]. Combining these three practices helps organizations better use their resources and save costs. The cloud allocates resources, agile prioritizes important features, and DevOps can maximize the return on investment by delivering value [4]. However, integrating these three methodologies for modern software development needs rigorous attention, as there is a lack of adequate research for modern software development. Therefore, to fill this deficiency, this research conducts a Systematic Literature Review (SLR) study, employing a structured approach to investigate the topic comprehensively. This study aims to address the crucial gap discussed above by inquiring into specific research questions mentioned in Table 1 to clarify the potential benefits and barriers associated with this integration.

* Corresponding author.
E-mail address: karima.moumane@ensias.um5.ac.ma (K. Moumane).

https://doi.org/10.1016/j.infsof.2024.107569
Received 9 May 2024; Received in revised form 31 July 2024; Accepted 29 August 2024
Available online 2 September 2024
0950-5849/© 2024 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Table 1
Research Questions (RQs) and Motivation.

Research Question (RQ)	Description
RQ1: What challenges and benefits arise from integrating Agile, Cloud, and DevOps in software development and how it impacts software testing processes?	To explore the challenges and benefits of combining Agile, Cloud, and DevOps, improving software development processes.
RQ2: How does mindset affect the successful merging of Agile, DevOps, and Cloud methodologies in software development?	To explore mindset's impact helps optimize the merging of Agile, DevOps, and Cloud for successful software development integration.
RQ3: How can mindset challenges be addressed for successful integration of Agile, DevOps, and Cloud methodologies in software development?	To ensure seamless integration of Agile, DevOps, and Cloud in software development, understanding and addressing mindset challenges are pivotal.
RQ4: Which Agile methodologies, particularly in their integration with DevOps and Cloud, are most commonly utilized?	To determine the most commonly used Agile methodologies in software development, especially when integrated with DevOps and Cloud.
RQ5: How does the integration of Agile, Cloud, and DevOps highlight the critical role of testing practices in improving software development efficiency and reliability?	Studying the testing emphasis within Agile, Cloud, and DevOps integration uncovers ways to improve software efficiency and reliability in modern development practices.
RQ6: What are the advantages and disadvantages of integrating Agile, cloud computing, and DevOps methodologies in the software development process?	This research question seeks to uncover both the advantages and drawbacks of merging Agile, cloud, and DevOps methodologies in software development.

The present research delves into the integration of DevOps and Agile methodologies with cloud infrastructure, examining their compatibility and the resulting challenges. Although Agile emphasizes iterative development and DevOps focuses on continuous delivery, integrating these with cloud technologies presents unique complexities. Challenges include aligning cloud-based resource management with Agile workflows and addressing the intricacies of cloud infrastructure within a DevOps pipeline. Currently, there is a lack of well-defined methodologies or processes for successful implementation and integration.

Our study aims to investigate the effective integration of Agile, DevOps, and cloud methodologies in challenging software development contexts. We will explore the impact of mindset on this integration, analyze the role of testing practices, and identify the most suitable Agile practices for seamless integration. The study seeks to provide valuable insights into the critical factors influencing the successful combination of these methodologies. These insights will not only bridge existing gaps in the literature but also offer practical guidance for organizations striving to enhance their software development practices through the effective integration of Agile, DevOps, and cloud technologies.

The structure of the paper is as follows: In Section 2, the background of the study is provided, shedding light on Agile methodologies, DevOps practices, and the integration of cloud infrastructure in software development. Section 3 outlines the proposed method used for this research, detailing how it is applied to achieve the study's main objectives. Following this, Section 4 presents the results obtained from the research work, providing insights into integrating DevOps, Agile, and cloud methodologies. Subsequently, Section 5 discusses the results' implications, exploring the challenges and opportunities associated with the integration process. Section 6 concludes the paper, summarizing the key findings. In Section 7, the validity of the obtained results is assessed, considering various factors such as reliability and generalizability. Finally, Sections Section 8,9 highlights avenues for future research and discuss the limitations of the study, respectively.

2. Background

The convergence of Agile, DevOps, and Cloud methodologies has drawn considerable attention in software development research. Studies have primarily aimed to uncover the advantages and hurdles associated with integrating these methodologies, to enhance the efficiency and effectiveness of software development processes.

Preliminary studies, such as [8], have been initiated to identify cloud-induced challenges and evaluate the effectiveness of Agile methodologies, notably Scrum and DevOps, in mitigating these challenges. Research has delved into the integration challenges when merging DevOps methodologies with cloud environments [9].

2.1. DevOps

DevOps is a collaborative approach to software development that emphasizes effective communication, seamless integration, and teamwork. It aims to facilitate cooperation between software development and operation teams, fostering a culture where collaboration is essential and every team member shares responsibilities while striving for continuous improvement. In addition to its core principles [10–12], DevOps encompasses several related concepts and practices that are crucial for understanding its implementation and impact. One such concept is DevSecOps, which integrates security practices into the DevOps process, ensuring that security integrates security practices into the DevOps process and that developers prioritize security throughout the software development lifecycle. Another vital framework is CALMS (Culture, Automation, Lean, Measurement, and Sharing), which stands for Culture, Automation, Lean, Measurement, and Sharing. CALMS provides a holistic view of the key components necessary for successful DevOps adoption, emphasizing cultural transformation, automation of repetitive tasks, lean principles to eliminate waste, measurement of performance metrics, and sharing of knowledge and experiences within the organization. Additionally, Continuous Integration and Continuous Deployment (CI/CD) are fundamental practices within the DevOps methodology, enabling frequent and automated testing, integration, and deployment of code changes, leading to faster delivery cycles and improved software quality.

2.2. Agile

Agile is a customer-oriented software development method that values adaptability, collaboration, and step-by-step progress [13]. It emphasizes delivering value to customers through constant feedback and evolution. Agile encourages teams to work closely together, communicate regularly, and respond quickly to changes in project requirements [14,15]. By promoting transparency and continuous improvement, Agile enables organizations to meet evolving customer needs and deliver high-quality software that matches user expectations. Agile project management is crucial for achieving successful project delivery by effectively managing risks in the face of dynamic business environments and evolving objectives [16]. The Agile methodology centers around flexibility, adaptability, and iterative development, requiring a particular mindset to ensure its successful implementation [17,18].

The most commonly used Agile methodologies include Kanban, Scrum, and Extreme Programming (XP):

Kanban: is an Agile framework that visualizes work as it moves through various process stages. It promotes continuous flow by limiting work in progress and focusing on completing tasks one at a time.

Scrum: is another Agile framework that divides work into small, manageable units called sprints. It emphasizes collaboration, frequent

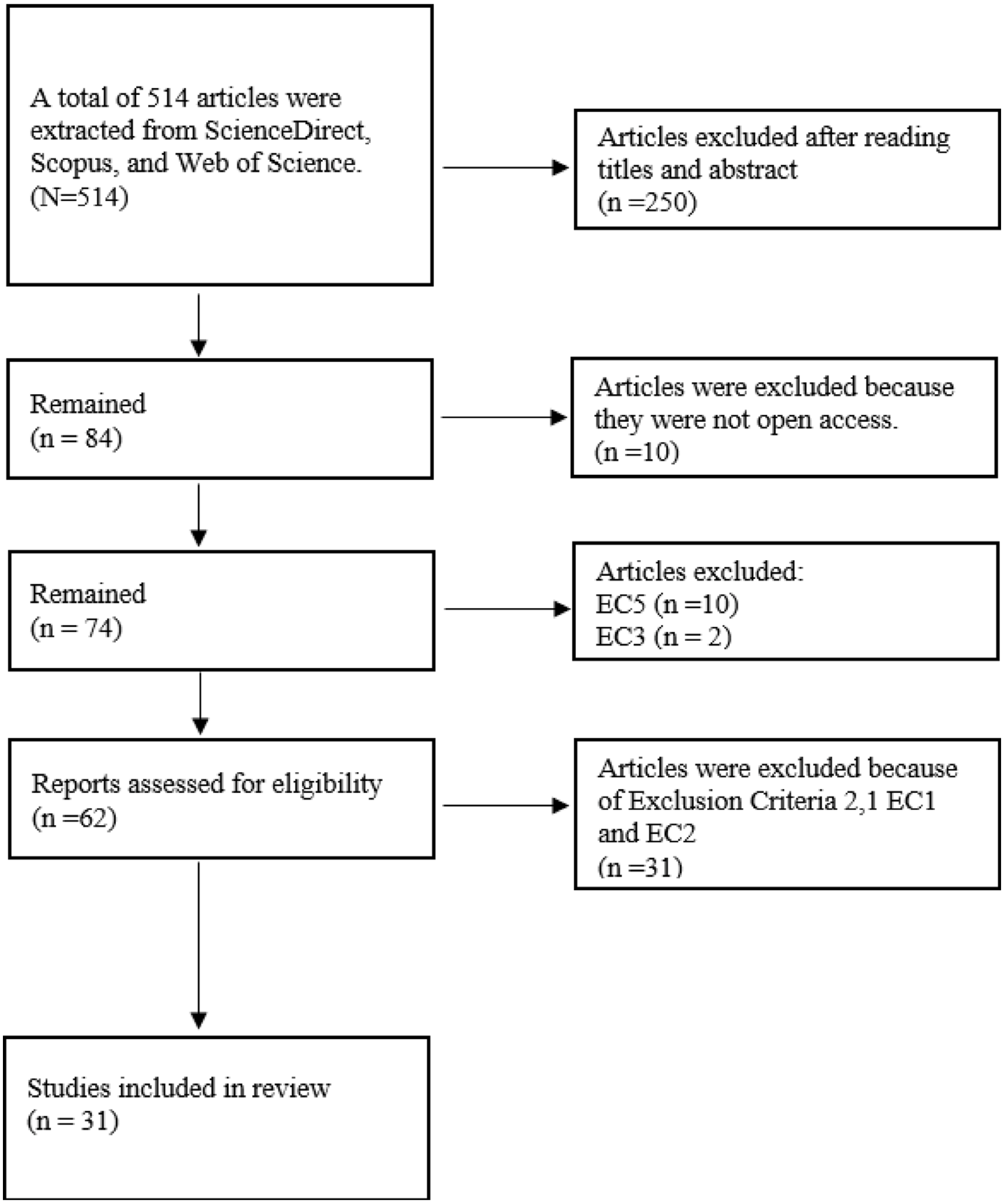

Fig. 1. PRISMA flow diagram.

inspection, and adaptation, with regular meetings such as daily stand-ups, sprint planning, and sprint reviews.

Extreme Programming (XP): is an Agile methodology focused on improving software quality and responsiveness to changing customer requirements. It advocates for pair programming, test-driven development, and continuous integration to achieve these goals.

2.3. Cloud

Cloud is a groundbreaking and omnipresent IT approach enabling immediate access to a flexible and shared reservoir of computing resources, including networks, servers, storage, applications, and services [19,20]. It facilitates resource provisioning and release in a dynamic, pay-as-you-go model, negating the need for organizations to invest in and maintain dedicated physical hardware [21]. Cloud offers inherent flexibility, scalability, and cost-efficiency, allowing users to rapidly scale resources up or down based on fluctuating demand, thereby optimizing resource utilization [22,23].

2.4. Software testing

Software testing is a systematic process for evaluating software applications' functionality, performance, and reliability. It involves assessing software components and functionalities to identify defects and ensure they meet specified requirements.

Unit Tests: These tests evaluate individual units or components of software to verify their correctness and functionality in isolation. They ensure that each unit operates as expected before integration into the more extensive system.

Tests of Performance and Scalability: These tests assess how software applications perform under varying workloads and stress conditions. They measure factors such as response time, data transfer rate, and resource usage to ensure optimal performance and scalability.

Non-Regression Tests: These tests verify that software modifications or updates do not introduce new defects or regressions. They ensure existing functionalities continue to work as intended, maintaining the stability and reliability of software systems over time.

3. Method

In this section, we describe our method for selecting articles. We used a structured approach called SLR to find and evaluate articles that match our research goals. We defined our research questions, listed the keywords we used for the search, explained our criteria for including or excluding articles, and detailed the sources we consulted.

3.1. Research questions (RQs)

The study defines a set of five research questions (RQs) to guide our research (Table 1). These RQs are designed to explore various aspects of integrating Agile, Cloud, and DevOps methodologies. They draw insights from the scientific literature and a preliminary analysis of the current landscape. These questions explore challenges, benefits, and the role of mindset in integration while proposing strategies to address mindset obstacles. Furthermore, they aim to identify commonly utilized Agile methodologies within DevOps and Cloud contexts. Overall, the goal is to gain insight into the collaborative functioning of these methods in modern software development.

3.2. Search terms and strategy

We began by identifying a set of keywords, considering abbreviations and synonyms. Our selected keywords encompassed terms such as "*DevOps*", "*Agile*", "*Cloud*", "*Mindset*", and "*comparison*". To ensure a comprehensive search, we employed logical connectors like "AND" and "OR" to combine these keywords effectively. This approach allowed us to cast a wide net, capturing relevant articles during our literature review process.

During the final step, we accessed the primary studies in databases to identify articles aligned with our search criteria. This process was executed by employing the following search string: *("Agile" AND ("DevOps" OR "CALMS") AND "Cloud" AND ("Comparison" OR "Differences" OR "setting-up" OR "Establishment" OR "Implementation") AND ("Culture" OR "Mindset"))* .

3.3. Search resources

To ensure a comprehensive review, we have utilized prominent databases: ScienceDirect, Scopus, and Web of Science. These choices are motivated by their wide use in the software domain, offering extensive resources and coverage, making them highly suitable for our research.

3.4. Exclusion and inclusion criteria

The definitive list of articles was generated using the specified exclusion and inclusion criteria.

Inclusion criteria were defined as follows:

1. English articles published from 2018.
2. Articles discussing the integration of DevOps, Agile, and cloud within software development, addressing their benefits and challenges.
3. Articles exploring similarities, differences, or implementation of these methods.
4. Articles study automation and continuous integration/continuous deployment (CI/CD) in architecture using both Agile and DevOps.

Exclusion criteria were defined as follows:

1. Articles that solely discuss single Agile, Cloud, or DevOps practices or tools without considering their broader integration in software development.
2. Articles addressing the impact of other technologies or practices on software development, excluding Agile and DevOps.
3. Non-open access articles.
4. Duplicate articles.

We researched articles published from 2018, sourcing data from databases including ScienceDirect, Scopus, and Web of Science. Initially, we identified 514 articles. After reviewing titles and abstracts, 250 articles were excluded. Following the removal of duplicates and non-English articles, 84 remained. Applying specific criteria, we excluded articles that did not align with our research question. This process resulted in a final selection of 31 articles, detailed in Table A.1 in the appendix and visually presented using a PRISMA diagram in Fig. 1.

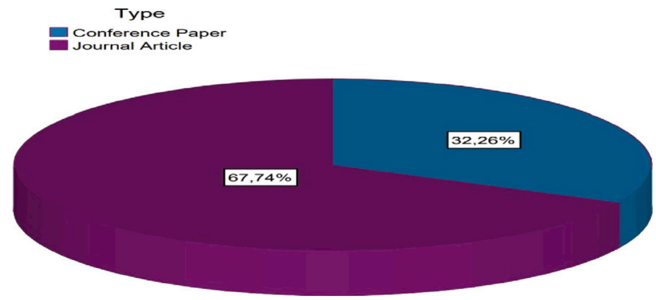

Fig. 2. Distribution of extracted articles by type.

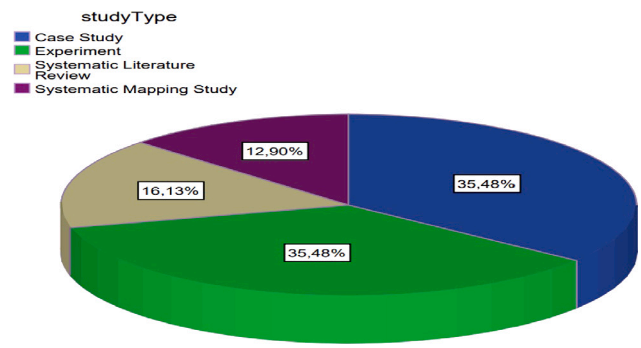

Fig. 3. Distribution of extracted articles by study type.

4. Results

This section shares insights from reviewing 31 selected articles in our SLR. We aim to extract valuable information and outcomes from analyzing and studying articles related to integrating Agile, DevOps, and Cloud approaches. Through this exploration, we seek to address the research questions outlined in Table 1.

A precise categorization emerges in the analyzed dataset comprising 31 extracted articles, as depicted in Fig. 2. The articles are divided into two primary types: conference papers and journal articles. According to the findings, 32% of the total articles fall under the classification of conference papers, while the remaining 68% are identified as journal articles.

From the 31 articles reviewed, the analysis distinguishes five articles as SLR, 11 as Experimental Studies, and 11 as Case Studies, with an additional four articles identified as systematic mapping studies as illustrated in Fig. 3. These findings underscore the diverse research methodologies adopted in the literature, highlighting case studies and experiments as the predominant approaches among the sampled articles. Fig. 4 presents the research design employed in the analyzed articles. The qualitative approach is prevalent, appearing in the majority of the articles, with a frequency of eleven times. Among these, seven

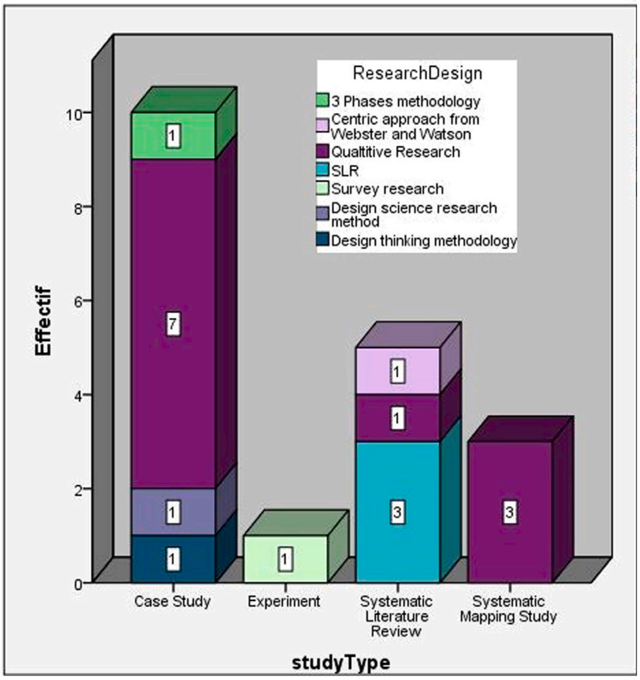

Fig. 4. Research approach by study type.

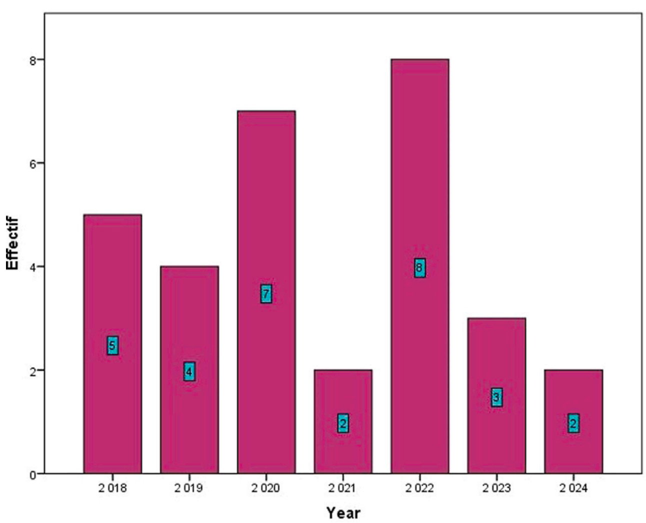

Fig. 5. The count of extracted articles over the years.

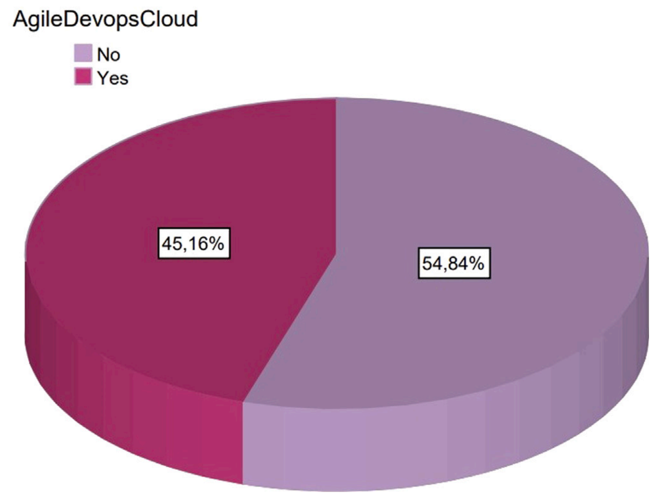

Fig. 6. Number of articles exploring the integration of the three approaches.

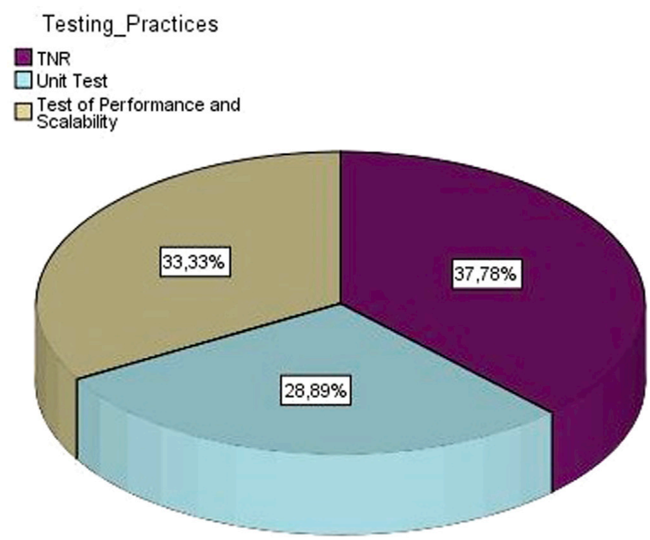

Fig. 7. Distribution of testing practices among articles utilizing test plan.

instances are attributed to the case study type, while three are for the systematic mapping study, and one instance pertains to the systematic literature review approach.

During our article extraction process, we considered articles published from 2018 onwards, as illustrated in Fig. 5. We excluded those published before this date. After reassessing this criterion, we found a distinctive distribution of articles across different years. Notably, 2022 had the highest number of articles, followed closely by 2020, which had seven more articles. In 2018, we identified five relevant articles, while the count for 2019 was four. Similarly, we included three articles from 2023 that adhered to our inclusion criteria. However, the years 2021 and 2024 had only two articles each.

The information extracted from Fig. 6 shows that out of the 31 scrutinized articles, only 14 explore or incorporate the integration of Agile, Cloud, and DevOps. This observation underscores the restricted representation of articles, as the majority of 17 articles fail to address these three approaches collectively, especially the technical aspects of integration.

In Fig. 7, examining articles employing plain tests unveils specific trends. Among these, 13 articles discuss or utilize Unit Tests, 15 articles concentrate on Tests of Performance and Scalability, and the majority 17 articles emphasize Non-Regression Tests (NRT). Some articles address multiple types of tests.

As shown in Fig. 8, within our corpus of extracted articles, a subset of sixteen articles extensively discuss or use the cloud. Among these sixteen articles, a particular observation has emerged from those that explicitly refer to cloud architecture. Specifically, the prevailing pattern indicates the adoption of microservices architecture in eight of the analyzed articles. This trend highlights a notable alignment between cloud infrastructure and the utilization of microservices architecture, suggesting an increasing recognition of the synergies between these two concepts.

In Fig. 9, we present the distribution of articles discussing or utilizing the three approaches: cloud, Agile, and DevOps. The categorization is based on the domain of the case study or experiment. Our analysis reveals that the IT domain is the most prevalent domain that embraces or discusses the integration of these three approaches. Among the nine

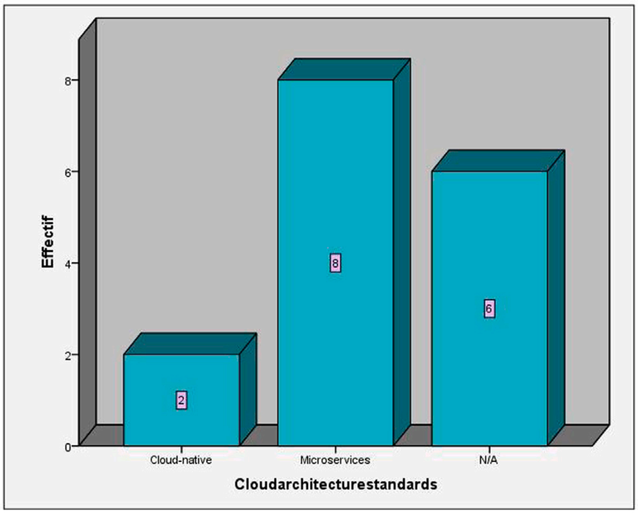

Fig. 8. Applied architectures within cloud environments.

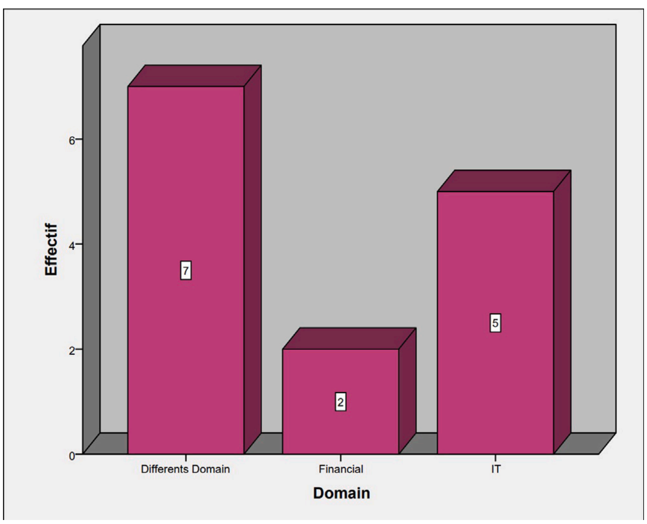

Fig. 9. Domain-wise distribution of articles.

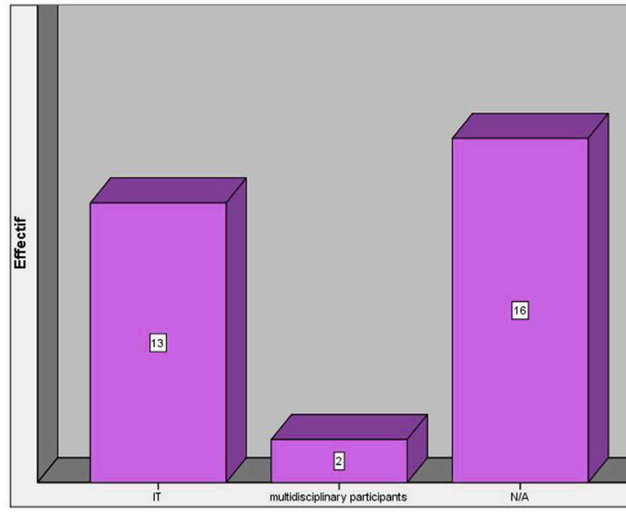

Fig. 10. Category of participants by study type.

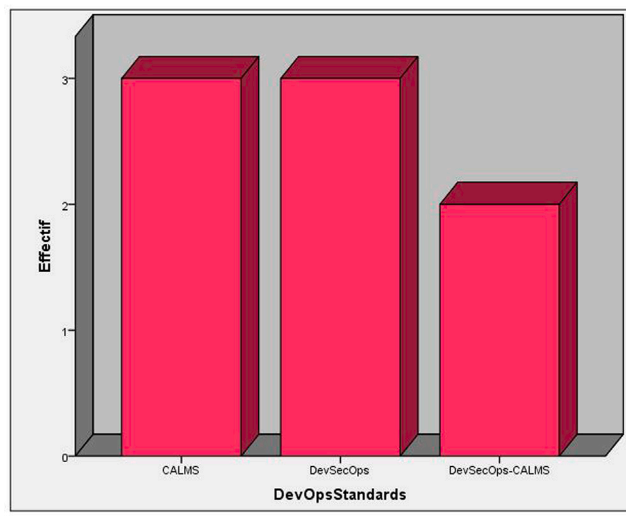

Fig. 11. DevOps standards adoption across articles.

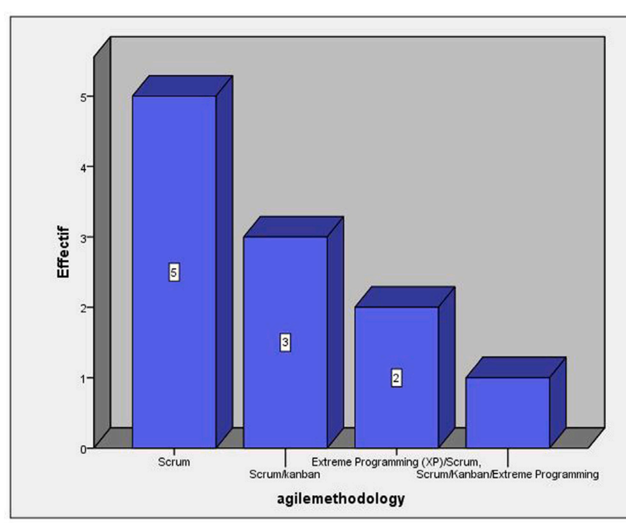

Fig. 12. Usage of agile methodologies in extracted articles.

articles that mention the domain, five specifically focus on the IT Domain.

The analysis of participants categorized by study type reveals interesting insights, as depicted in Fig. 10. It is evident from the graph that the dominant categories of participants in both case studies and experiments belong to the IT domain. Specifically, participants from the IT sector, encompassing developers and engineers from various subdomains, are prominently represented across different graph segments.

Fig. 11 visually captures the focus of DevOps-related articles, revealing a distinct emphasis on the utilization of DevOps standards. Notably, the figure highlights that the most frequently referenced standards within these discussions are DevSecOps and CALMS. This pattern underscores the industry's recognition of the importance of security integration (DevSecOps) and comprehensive guideline adherence (CALMS) in successful DevOps endeavors. Based on the graph in Fig. 12, our analysis focused on identifying the presence of Agile methodologies in the articles we extracted. The criterion for this identification was based on whether an article utilized or discussed Agile methodologies. Our findings revealed that Scrum emerged as the most prominent methodology among the articles mentioning or using Agile methods. It was consistently present in all articles that discussed or utilized Agile. Moreover, Scrum was found to be employed either independently or

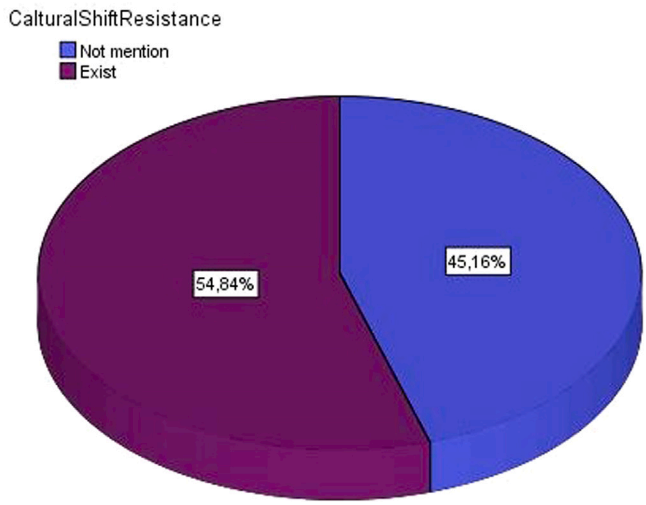

Fig. 13. Counting mentions of cultural shift across articles.

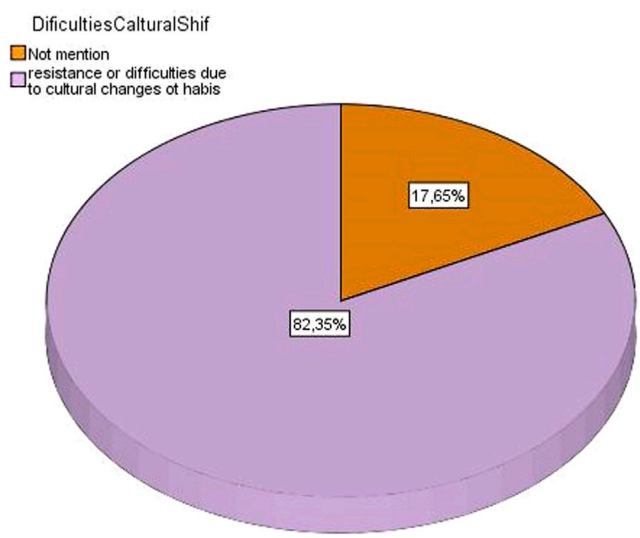

Fig. 14. Challenges arising from cultural shift.

in combination with other technologies, such as Kanban or Extreme Programming, across all the relevant articles.

In Fig. 13, depicting an analysis of 31 extracted articles, it becomes apparent that 17 articles focus on cultural shift issues. This subset of articles highlights and underscores cultural shift challenges' substantial presence and significance, specifically within integrating DevOps, cloud, and agile approaches. It emphasizes the critical role that cultural transformations play in successfully implementing and fusing these methodologies within contemporary organizational frameworks.

As outlined in the data presented in Fig. 14, among the 17 articles that discuss cultural shift resistance, a significant majority of 14 articles specifically highlight the challenges associated with resistance or difficulties arising from cultural changes or habitual shifts.

The pivotal solutions outlined in Fig. 15 have been distilled from an extensive analysis encompassing 17 articles, as elucidated in Fig. 13. These key strategies are instrumental for accomplishing a successful cultural shift in integrating DevOps, Cloud, and Agile methodologies. The allocation percentages for each solution are as follows: Prioritizing Documentation and Traceability (20%), Emphasis on Collaboration and Communication (13.33%), DevOps Cultural Integration (40%), Emphasizing Culture over Tools (13.33%), and Implementing DevOps Guidelines (13.33%).

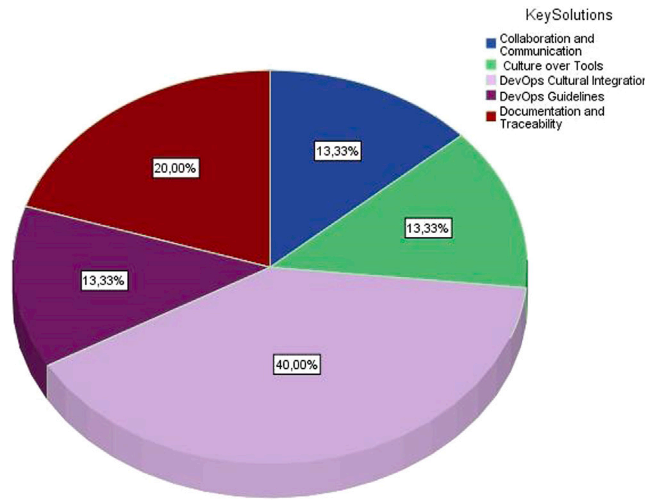

Fig. 15. Key solutions extracted for cultural shift.

5. Discussion

Comprising 31 articles gathered since 2018(Fig. 5, the dataset is categorized into conference papers and journal articles, as depicted in Fig. 2, underscoring the importance of Agile, DevOps, and Cloud in software development. Conference papers prioritize practical applications, providing tangible insights, while journal articles delve into more profound analyses. Fig. 3 additionally emphasizes a preference for practical study types, with a higher count of Experimental Studies and Case Studies, accentuating the pragmatic nature of these methodologies in real-world scenarios.

As seen in Fig. 4, when we dive deeper into the details, it is clear that the qualitative method is primarily expressed through the use of case studies. This approach is evident in seven out of the ten instances where qualitative methods are used. The qualitative approach offers several research advantages. It provides deep insights into complex phenomena through case studies and interviews, capturing nuances that quantitative methods may miss. This approach delves into the 'why' and 'how' aspects, making it suitable for exploring subjective experiences and enhancing our understanding of human behavior. Its adaptability allows for refining questions and hypotheses during the study. In summary, qualitative research uncovers deeper meanings, promotes a comprehensive understanding, and generates hypotheses for further exploration.[24].

Our research has uncovered valuable insights into the integration of ARQ1: What challenges and benefits arise from integrating Agile, Cloud, and DevOps in software development, and how does this overview of our findings, including challenges, benefits, and the nuances related to mindset. Furthermore, we explore strategies to address these mindset challenges effectively, ensuring a seamless and flawless integration process. Additionally, we extract standard Agile methodologies employed in this integration, underscoring the significance of rigorous testing. This discussion illuminates the present state of software development employing these integrated approaches.

RQ1:What challenges and benefits arise from integrating Agile, Cloud, and DevOps in software development and how it impact software testing processes?

Addressing RQ1, our analysis of the 31 extracted articles reveals that 14 articles explicitly discuss or implement the three approaches Agile, Cloud, and DevOps (Fig. 6). Conversely, the remaining 17 articles manifest a lack of discourse or utilization of these approaches, signaling potential challenges for researchers and practitioners when applying Agile, Cloud, and DevOps in their software projects. It is noteworthy that all the extracted articles belong to the IT domain, with the predominant category of participants in case studies and experiments being IT professionals (Figs. 9,10). These challenges may stem from the intricacies of implementing each approach's associated methodologies and tools, especially since tools play a crucial role in facilitating the implementation and success of practices [25]. The presence of numerous tools specific to each methodology can introduce complexities in integration. However, opting for tools capable of monitoring and integrating multiple approaches, such as Jenkins, GitLab, Kubernetes, and Docker, may pave the way for more successful outcomes [26]. Overall, 14 articles discussing the integration of Agile, Cloud, and DevOps demonstrate the growing interest and recognition of their combined potential. It underscores the need for continuous research and practical implementation to exploit the advantages of these methodologies together. Combining Agile, Cloud, and DevOps offers significant benefits for software development. It improves collaboration, speeds up development, and simplifies deployment. Agile ensures better customer focus, the cloud provides scalable resources, and DevOps automates processes. This integration results in faster innovation, higher quality products, and a competitive advantage in the fast-changing tech world [4,27,28].

RQ2:How does mindset affect the successful merging of Agile, DevOps, and Cloud methodologies in software development?

The successful integration of Agile, DevOps, and Cloud methodologies within software development is significantly influenced by mindset. As depicted in Fig. 13, an analysis of 31 articles reveals that challenges related to cultural shifts present a substantial hurdle in this integration, with 14 articles specifically emphasizing these challenges. Furthermore, among the 17 articles discussing resistance to cultural shifts, Fig. 14 highlights that a significant majority, comprising 14 articles, address the obstacles linked to resistance or difficulties stemming from cultural changes or habitual shifts. These findings underscore the pivotal role of mindset in navigating cultural transformations, where an adaptable and open mindset emerges as crucial [29].

Moreover, examining resistance to cultural shifts in adopting DevOps practices underscores the significance of mindset in this context. The DevOps philosophy stands out as a crucial component in understanding the heightened impact of mindset on successful integration within the software development landscape. DevOps, not just a set of technical practices [30–33], encompasses a broader set of cultural values and principles. Its primary aim is to seamlessly integrate the software industry, break down barriers, and notably enhance collaboration and communication in a highly effective manner. In considering the amalgamation of Agile, DevOps, and Cloud methodologies, the role of mindset, especially within the DevOps paradigm, becomes remarkably pronounced [34].

This essential mindset involves acknowledging change as a natural and beneficial part of the development process. It fosters a collaborative environment, emphasizing transparent communication, teamwork, and collaboration among various stakeholders, cross-functional teams, and customers [35]. Moreover, an Agile mindset prioritizes customer-centricity by aligning development initiatives with evolving customer requirements and desires. It flourishes through incremental progress, promoting ongoing enhancements and consistently delivering value at every developmental phase [36].

The successful integration of Agile, DevOps, and Cloud methodologies within software development hinges on a mindset that values adaptability, collaboration, and customer-centricity. DevOps, extending beyond technical practices, emphasizes cultural values aimed at seamless integration and enhanced collaboration. Meanwhile, Agile methodology prioritizes iterative progress, transparent communication, and customer satisfaction. Both methodologies underscore the pivotal role of mindset in navigating change and delivering value at every developmental stage.

RQ3: How can mindset challenges be addressed for successful integration of Agile, DevOps, and Cloud methodologies in software development?

Addressing the challenges associated with mindset is pivotal for successfully integrating Agile, DevOps, and Cloud methodologies within software development. The delineation of five key solutions in Fig. 15 consolidates foundational strategies extracted from comprehensive studies, providing a roadmap to navigate and mitigate these challenges effectively.

Firstly, prioritizing documentation and traceability emphasizes the importance of reliable information to foster collaboration among technical staff and project leaders. Implementing tools such as Confluence, Jira, GitLab or GitHub, Microsoft SharePoint, Google Workspace, or Wiki Systems significantly aids in achieving this objective. Moreover, the focus on collaboration and communication advocates for a culture of cooperation, amplifying efficiency and aligning technological endeavors with broader business goals. Adopting platforms like Slack, Microsoft Teams, Zoom, Trello, Asana, Google Workspace, Miro, or Monday.com significantly strengthens the emphasis on Collaboration and Communication, nurturing a cooperative culture, enhancing efficiency, and aligning technological initiatives with broader business objectives [37,38].

DevOps cultural integration serves as a bridge to overcome communication gaps and cultivates an atmosphere of cross-functional teamwork. This integration approach aligns closely with the cultural values advocated by frameworks like CALMS (Fig. 11) and the DevOps Institute's SKIL Framework. These frameworks emphasize the importance of cultural aspects alongside technical elements, recognizing the significance of architecture and frameworks is crucial not only for DevOps but also in the context of cloud integration(Fig. 8). Moreover, practices such as cross-functional collaboration, continuous feedback loops, and leadership support are often cited as effective methods to ensure DevOps Cultural Integration. Implementing DevOps Guidelines revolves around restructuring, training, and aligning teamwork. Viewing DevOps as a Cultural Shift emphasizes collaboration, automation, and quality measurement, whereas Challenges of Innovation address resistance to new technologies. Additionally, the stress on Emphasizing Culture over Tools highlights the necessity for a cultural transformation rather than mere tool adoption, which is essential for holistic integration.

These comprehensive solutions, derived from various studies, form a structured approach to addressing mindset challenges, creating an environment conducive to seamlessly integrating Agile, DevOps, and Cloud methodologies in software development.

RQ4: Which Agile methodologies, particularly in their integration with DevOps and Cloud, are most commonly utilized?

The analysis findings (Fig. 12) reveal Scrum's dominance as the primary Agile methodology. Every discussion or employment of Agile methodologies underscores the prominence of Scrum. This consistent emergence of Scrum, both as a standalone practice and in tandem with technologies like Kanban or Extreme Programming, highlights its vital role in Agile practices [39]. This underscores its wide acceptance and adaptability within the studied domain. The data reinforces Scrum's stature as a trusted framework for Agile methodologies, demonstrating flexibility when used independently or collaboratively with other methods. This adaptability is especially significant in facilitating effective project management and development processes. The success of Scrum depends on the team's maturity level when implementing it.[40]. Teams with a higher maturity level are better equipped to leverage Scrum's potential, aligning with DevOps principles to promote collaboration, continuous improvement, and streamlined workflows. This connection emphasizes that Scrum's effectiveness extends beyond Agile principles and is crucial in facilitating the DevOps culture and practices. Teams with a mature understanding of Scrum and DevOps can seamlessly integrate their methodologies, leading to enhanced project management, development processes, and overall organizational efficiency.

RQ5:How does the integration of Agile, Cloud, and DevOps highlight the critical role of testing practices in improving software development efficiency and reliability?

Among the 14 articles that discuss or apply Agile, Cloud, and DevOps integration, 11 of them employ plain tests(Fig. 7). This emphasis on plain tests offers valuable insights into the testing practices of researchers and practitioners. Notably, these articles have identified three distinct types of tests: Unit Tests, Tests of Performance and Scalability, and Non Regression Tests. Unit Tests, which are discussed or utilized in 13 of the articles, play a critical role in software development by assessing individual units or components in isolation, verifying the correctness and functionality of small code segments, ensuring that each unit operates as expected before integration into the larger system [41,42]. The emphasis on Unit Tests reflects the significance of Non-regression tests as national building blocks of software solutions. Additionally, unit tests act as documentation for the codebase, enhancing software maintainability and facilitating smooth updates and modifications over time [43,44]. Another prominent category is the Test of Performance and Scalability, highlighted in 15 articles. These tests assess how software applications perform under varying workloads and stress conditions, measuring response time, data transfer rate, and resource usage. Performance and scalability are critical considerations, especially in modern applications with high user expectations and growing demands [45,46]. Notably, Non-Regression Tests occupy the most significant portion, with 17 articles focusing on this type. Non-Regression Tests are instrumental in verifying that software modifications or updates do not introduce new defects or regressions, ensuring that existing functionalities continue to work as intended [47,48]. This emphasis on Non Regression Tests underscores the importance of maintaining the stability and reliability of software systems over time, even as new features and changes are introduced [49]. By studying the distribution of testing practices in articles utilizing plain tests, we gain valuable insights into the testing priorities and trends in software development research and practices. In an Agile and DevOps environment, the rapid development and continuous delivery require frequent execution of Unit Tests to ensure that individual components function correctly before integration. The cloud provides the necessary resources for scaling performance and scalability tests as required, aligning with DevOps' focus on automation and efficiency. Additionally, DevOps practices streamline the execution of Non-Regression Tests, as they often involve verifying that software modifications do not introduce defects or regressions. This process benefits greatly from automation. This merger enhances the efficiency and effectiveness of these crucial testing methodologies, promoting quicker and more dependable software development.

In summary, the results and insights presented in this study are derived from an analysis of 31 articles [4,8,9,28,30–33,50–72] . These findings shed light on the challenges and benefits of integrating Agile, Cloud, and DevOps methodologies in software development, highlighting the pivotal role of mindset in navigating cultural transformations, the prominence of Scrum as the primary Agile methodology, and the critical importance of testing practices in enhancing software development efficiency and reliability.

6. Conclusion

In conclusion, our analysis of 31 articles on integrating Agile, Cloud, and DevOps in software development highlights a discernible interest and acknowledgment of their combined potential. Challenges in implementing associated methodologies and tools are evident, but adopting tools like Jenkins, GitLab, Kubernetes, and Docker holds promise in overcoming these complexities. Integrating Agile, Cloud, and DevOps offers substantial benefits, enhancing collaboration, expediting development, and simplifying deployment processes. Scrum is the predominant Agile methodology, underscoring its adaptability within the studied domain. Mindset is crucial in navigating cultural shifts for successful integration, and addressing mindset challenges is pivotal for creating an environment conducive to seamless integration.

Testing practices, including Unit Tests, Performance and Scalability Tests, and Non Regression Tests, emerge as critical components in improving software development efficiency and reliability within the integrated framework. These testing methodologies are pivotal in an Agile and DevOps environment, contributing to quicker and more dependable software development. The landscape of contemporary software development is thus characterized by a dynamic interplay of methodologies, tools, and cultural considerations, emphasizing the need for ongoing research and practical implementation to exploit the advantages of this integrated approach fully.

Our comprehensive analysis provides a nuanced understanding of the challenges, benefits, prevalent methodologies, and the critical role of testing practices in integrating Agile, Cloud, and DevOps. This exploration contributes valuable insights into the complex intersection of these methodologies, providing a foundation for continued research and practical applications in the ever-evolving landscape of software development.

RQ6: What are the advantages and disadvantages of integrating Agile, cloud computing, and DevOps methodologies in the software development process?

In addressing RQ6, we delve into the advantages and disadvantages of integrating Agile, cloud, and DevOps methodologies in software development. Our analysis reveals several benefits of this integration, including improved collaboration between development and operations teams, faster development cycles, and enhanced scalability and flexibility cloud infrastructure offers. The iterative nature of Agile methodologies enables teams to adapt quickly to changing requirements. At the same time, DevOps practices promote automation and continuous delivery, resulting in faster time-to-market and improved overall efficiency. Moreover, using cloud resources provides access to scalable infrastructure on-demand, reducing operational costs and allowing for rapid scalability to meet fluctuating demand.

However, despite these benefits, several challenges accompany the integration of Agile, cloud, and DevOps methodologies. These include the complexity of managing multiple tools and technologies, as mentioned in RQ1, cultural resistance to change within organizations, and potential security and compliance issues associated with cloud-based deployments. Adopting new practices and technologies may require significant training and organizational restructuring, leading to initial productivity disruptions. Additionally, reliance on cloud infrastructure introduces dependencies on external service providers, raising concerns about data privacy, reliability, and vendor lock-in.

After conducting a detailed analysis of the reviewed articles, we have gained valuable insights into the intersection of Agile, Cloud, and DevOps in software development. We found a significant difference between the 14 articles that incorporate these three approaches and the 17 that do not. This highlights the challenges and complexities researchers and practitioners face when adopting a holistic approach.

7. Threats to validity

There are some threats to the validity of our study. They are described and detailed as follows:

- Selection Bias: There might be a risk of selection bias in the literature included in our SLR, as we rely on the availability and accessibility of published articles and may overlook relevant sources that are not easily accessible.
- Methodological Limitations: Our study's findings may be influenced by methodological limitations inherent in SLR, such as variability in search strategies, inclusion criteria, and data extraction processes, which could impact the comprehensiveness and reliability of our results.
- Generalizability: The findings of our study may be limited in their generalizability to different contexts or industries due to variations in organizational structures, cultural norms, and technological landscapes.

Table A.1
Data Items.

Reference	Year	Title
[50]	2018	Managing Agile and Dev-Ops Activities Through a Consolidated Web Portal: South African Case Study
[4]	2022	Exploring the Benefits of Combining DevOps and Agile
[51]	2018	A DevOps Implementation Framework for Large Agile-Based Financial Organizations
[9]	2019	Major Challenges of Systems-of-Systems with Cloud and DevOps - a financial experience report
[52]	2019	Transformation of Consulting for SoftwareDefined Businesses: Lessons from a DevOps Case Study in a German IT Company
[53]	2019	Implementing Large Enterprise Resource Planning Systems with Agile Methods
[54]	2023	DevOps critical success factors : A systematic literature review
[55]	2022	Implement the already proposed DevOps hybrid model using suggested tool
[56]	2021	Security in agile software development: A practitioner survey
[57]	2020	DevOps and software quality: A systematic mapping
[58]	2020	DevOps in Practice for Education Management Information System at ECNU
[59]	2023	Framework and tooling proposals for Agile certification of safety-critical embedded software in avionic systems
[60]	2022	A Cloud Computing web-based application for Smart Farming based on microservices architecture
[61]	2019	Adopting DevOps in the real world: A theory, a model, and a case study
[62]	2020	Measurement Based Performance Evaluation of DevOps
[30]	2022	Exploration of DevOps testing process capabilities: An ISM and fuzzy TOPSIS analysis developed a conceptual framework to improve DevOps testing process
[28]	2022	Automated Configuration for Agile Software Environments
[63]	2020	An Empirical Taxonomy of DevOps in Practice
[64]	2018	IT Governance Mechanisms for DevOps Teams : How Incumbent Companies Achieve Competitive Advantages
[65]	2020	A Systematic Literature Review on DevOps Capabilities and Areas
[31]	2022	Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic Review
[32]	2018	A case analysis of enabling continuous software deployment through knowledge management
[66]	2022	A mapping study on documentation in Continuous Software Development
[67]	2020	A Systematic Mapping Study on Microservices Architecture in DevOps
[68]	2024	Architectural support for software performance in continuous software engineering: A systematic mapping study
[69]	2018	Automation of Regression test in Microservice Architecture
[70]	2022	Crane Cloud: A resilient multi-cloud service abstraction layer for resource-constrained settings
[33]	2024	Is it worth adopting DevOps practices in Global Software Engineering? Possible challenges and benefits
[71]	2020	Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)
[72]	2023	Toward an integrated framework for developing European 6G innovation
[8]	2021	Determining the Benefits and Drawbacks of Agile (Scrum) and DevOps in Addressing the Development Challenges of Cloud Applications

- Time Constraints: Given the dynamic nature of the software development field, our study's findings may be subject to time constraints as new methodologies, tools, and practices continue to emerge after the completion of our literature review.

8. Future works

Our study's findings provide opportunities for future exploration, especially in contrasting the impact of mindset and tools on effectively integrating Agile, Cloud, and DevOps methodologies. A more in-depth analysis of the challenges associated with mindset shifts and the corresponding strategies for overcoming resistance could provide valuable insights. Understanding how different organizational cultures and individual mindsets influence the adoption and effectiveness of these methodologies is essential for fostering successful integration.

On a parallel note, the role of tools in facilitating the integration process warrants closer attention. The effectiveness of specific tools like Jenkins, GitLab, Kubernetes, and Docker in addressing challenges and streamlining processes could offer practical insights for practitioners and researchers. Additionally, evaluating the compatibility and synergy among these tools within an integrated framework can contribute to optimizing the software development workflow.

Moreover, as our technological landscape continues to evolve, investigating the intersection of Agile, Cloud, and DevOps with emerging trends like serverless computing, edge computing, and artificial intelligence becomes crucial. How do these trends complement or challenge the existing integrated approaches? Exploring these intersections can guide the adaptation of methodologies to stay relevant and effective in the face of technological advancements.

In essence, future research avenues should not only delve into the human aspect of mindset but also rigorously explore the role of tools in achieving successful integration. By contrasting these elements, researchers can provide comprehensive insights into how organizations can navigate challenges and leverage opportunities in the dynamic field of software development.

9. Study limitations

Our study has a few limitations that should be considered when interpreting the findings. Firstly, the systematic literature review only includes publications available up to 2024, which may not capture the latest advancements or emerging trends in the integration of Agile, Cloud, and DevOps technologies. Additionally, our dataset is limited to open-access articles, potentially excluding valuable insights from studies published in subscription-based journals, which could impact the comprehensiveness and depth of the review. While our study highlights the significance of tools such as Jenkins, GitLab, Kubernetes, and Docker in integration, it is essential to recognize that the analysis is based on existing literature, and practical insights from real-world implementations of these tools may vary. This calls for further empirical research to validate their effectiveness across diverse scenarios. Furthermore, although our study emphasizes the importance of mindset and organizational culture in the integration process, it is important to acknowledge that the depth of analysis on these human factors is limited by the availability of detailed case studies and qualitative data in the reviewed literature. A more extensive qualitative research approach could provide deeper insights into these aspects.

CRediT authorship contribution statement

Fatiha El Aouni: Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Data curation, Conceptualization. **Karima Moumane:** Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Data curation, Conceptualization. **Ali Idri:** Writing – review & editing. **Mehdi Najib:** Writing – review & editing, Validation. **Saeed Ullah Jan:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix

See Table A.1

References

[1] Sochova, Zuzana, Software development methodology for fast changing environment, in: New Trends in Software Methodologies, Tools and Techniques, IOS Press, 2009, pp. 379–388, http://dx.doi.org/10.3233/978-1-60750-049-0-379, URL https://ebooks.iospress.nl/doi/10.3233/978-1-60750-049-0-379.

[2] Eoin Woods, Software architecture in a changing world, IEEE Softw. (ISSN: 1937-4194) 33 (6) (2016) 94–97, http://dx.doi.org/10.1109/MS.2016.149, Conference Name: IEEE Software, URL https://ieeexplore.ieee.org/abstract/document/7725217/references#references.

[3] Lianping Chen, Continuous delivery: Overcoming adoption challenges, J. Syst. Softw. (ISSN: 0164-1212) 128 (2017) 72–86, http://dx.doi.org/10.1016/j.jss.2017.02.013, URL https://www.sciencedirect.com/science/article/pii/S0164121217300353.

[4] Fernando Almeida, Jorge Simoes, Sergio Lopes, Exploring the benefits of combining DevOps and agile, Future Internet (ISSN: 1999-5903) 14 (2) (2022) 63, http://dx.doi.org/10.3390/fi14020063, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, URL https://www.mdpi.com/1999-5903/14/2/63.

[5] Mohammed Airaj, Enable cloud DevOps approach for industry and higher education, Concurr. Comput.: Pract. Exper. (ISSN: 1532-0634) 29 (5) (2017) e3937, http://dx.doi.org/10.1002/cpe.3937, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.3937. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3937.

[6] Dhaya Sindhu Battina, Devops, a new approach to cloud development & testing, 2020, URL https://papers.ssrn.com/abstract=4004330.

[7] Sowmya Karunakaran, Impact of cloud adoption on agile software development, in: Zaigham Mahmood, Saqib Saeed (Eds.), Software Engineering Frameworks for the Cloud Computing Paradigm, in: Computer Communications and Networks, Springer, London, ISBN: 978-1-4471-5031-2, 2013, pp. 213–234, http://dx.doi.org/10.1007/978-1-4471-5031-2_10.

[8] Konstantinos Tsilionis, Sarah Sassenus, Yves Wautelet, Determining the benefits and drawbacks of agile (scrum) and DevOps in addressing the development challenges of cloud applications, in: Anna Visvizi, Orlando Troisi, Kawther Saeedi (Eds.), Research and Innovation Forum 2021, in: Springer Proceedings in Complexity, Springer International Publishing, Cham, ISBN: 978-3-030-84311-3, 2021, pp. 109–123, http://dx.doi.org/10.1007/978-3-030-84311-3_11.

[9] S.B.O.G. Caraturan, D.H. Goya, Major challenges of systems-of-systems with cloud and DevOps - a financial experience report, ISBN: 978-1-72813-439-0, 2019, pp. 10–17, http://dx.doi.org/10.1109/SESoS/WDES.2019.00010.

[10] Liming Zhu, Len Bass, George Champlin-Scharff, DevOps and its practices, IEEE Softw. (ISSN: 1937-4194) 33 (3) (2016) 32–34, http://dx.doi.org/10.1109/MS.2016.81, Conference Name: IEEE Software.

[11] Mayank Gokarna, Raju Singh, DevOps: A historical review and future works, in: 2021 International Conference on Computing, Communication, and Intelligent Systems, ICCCIS, 2021, pp. 366–371, http://dx.doi.org/10.1109/ICCCIS51004.2021.9397235.

[12] Nicole Forsgren, Mik Kersten, DevOps metrics, Commun. ACM (ISSN: 0001-0782) 61 (4) (2018) 44–48, http://dx.doi.org/10.1145/3159169.

[13] Peggy Gregory, Katie Taylor, Defining agile culture: A collaborative and practitioner-led approach, in: 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE, 2019, pp. 37–38, http://dx.doi.org/10.1109/CHASE.2019.00016, ISSN: 2574-1837.

[14] Darja Smite, Nils Brede Moe, Javier Gonzalez-Huerta, Overcoming cultural barriers to being agile in distributed teams, Inf. Softw. Technol. (ISSN: 0950-5849) 138 (2021) 106612, http://dx.doi.org/10.1016/j.infsof.2021.106612, URL https://www.sciencedirect.com/science/article/pii/S0950584921000884.

[15] Kashumi Madampe, Rashina Hoda, John Grundy, Paramvir Singh, Towards understanding technical responses to requirements changes in agile teams, in: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW '20, Association for Computing Machinery, New York, NY, USA, ISBN: 978-1-4503-7963-2, 2020, pp. 153–156, http://dx.doi.org/10.1145/3387940.3392229.

[16] Katarina Buganova, Jana Simickova, Risk management in traditional and agile project management, Transp. Res. Procedia (ISSN: 2352-1465) 40 (2019) 986–993, http://dx.doi.org/10.1016/j.trpro.2019.07.138, URL https://www.sciencedirect.com/science/article/pii/S2352146519303060.

[17] Laurens van Driessen, Agile implementation in hardware environment, 2023, URL https://repository.tudelft.nl/islandora/object/uuid%3A1e4c276b-4767-449e-a7a3-323329685085.

[18] Successful agile implementation through the agility mindset and digital intelligence - University of Johannesburg, URL https://ujcontent.uj.ac.za/esploro/outputs/graduate/Successful-agile-implementation-through-the-agility/9933809907691.

[19] Mohammad Ilyas Malik, CLOUD COMPUTING-technologies, Int. J. Adv. Res. Comput. Sci. 9 (2018) 379–384, http://dx.doi.org/10.26483/ijarcs.v9i2.5760.

[20] Larry Combs, Cloud computing for SCADA, Control Eng. (ISSN: 00108049) 58 (12) (2011) 22–26, Publisher: CFE Media LLC, URL https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=00108049&v=2.1&it=r&id=GALE%7CA345882699&sid=googleScholar&linkaccess=abs.

[21] Isaac Odun-Ayo, M. Ananya, Frank Agono, Rowland Goddy-Worlu, Cloud computing architecture: A critical analysis, in: 2018 18th International Conference on Computational Science and Applications, ICCSA, 2018, pp. 1–7, http://dx.doi.org/10.1109/ICCSA.2018.8439638.

[22] Priyanshu Srivastava, Rizwan Khan, A review paper on cloud computing, Int. J. Adv. Res. Comput. Sci. Softw. Eng. 8 (2018) 17, http://dx.doi.org/10.23956/ijarcsse.v8i6.711.

[23] Aaqib Rashid, Amit Chaturvedi, Cloud computing characteristics and services: A brief review, Int. J. Comput. Sci. Eng. 7 (2019) 421–426, http://dx.doi.org/10.26438/ijcse/v7i2.421426.

[24] David Silverman, Qualitative research, theory, method and practice, 1998.

[25] Sergio Galvan-Cruz, Mirna Munoz, Jezreel Mejia, Open-source tools and their coverage to implement DevOps, in: Jezreel Mejia, Mirna Munoz, Alvaro Rocha, Yasmin Hernandez Perez, Himer Avila-George (Eds.), New Perspectives in Software Engineering, Springer Nature Switzerland, Cham, ISBN: 978-3-031-50590-4, 2024, pp. 17–38, http://dx.doi.org/10.1007/978-3-031-50590-4_2.

[26] Charanjot Singh, Nikita Seth Gaba, Manjot Kaur, Bhavleen Kaur, Comparison of different CI/CD tools integrated with cloud platform, in: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 7–12, http://dx.doi.org/10.1109/CONFLUENCE.2019.8776985.

[27] Sara Carturan, Denise Goya, Major challenges of systems-of-systems with cloud and DevOps – a financial experience report, in: 2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, WDES, 2019, pp. 10–17, http://dx.doi.org/10.1109/SESoS/WDES.2019.00010.

[28] Negar Mohammadi Koushki, Sanjeev Sondur, Krishna Kant, Automated configuration for agile software environments, in: 2022 IEEE 15th International Conference on Cloud Computing, CLOUD, 2022, pp. 511–521, http://dx.doi.org/10.1109/CLOUD55607.2022.00074, ISSN: 2159-6190.

[29] Muhammad Yazid Al Qahar, Teguh Raharjo, DevOps implementation challenges in the Indonesian public health organization, Int. J. Adv. Comput. Sci. Appl. (IJACSA) (ISSN: 2156-5570) 14 (9) (2023) http://dx.doi.org/10.14569/IJACSA.2023.0140910, Number: 9 Publisher: The Science and Information (SAI) Organization Limited, URL https://thesai.org/Publications/ViewPaper?Volume=14&Issue=9&Code=IJACSA&SerialNo=10.

[30] Saima Rafi, Muhammad Azeem Akbar, Wu Yu, Ahmed Alsanad, Abdu Gumaei, Muhammad Sarwar, Exploration of DevOps testing process capabilities: An ISM and fuzzy TOPSIS analysis, Appl. Soft Comput. 116 (2021) 108377, http://dx.doi.org/10.1016/j.asoc.2021.108377.

[31] Muhammad Khan, Abdul Khan, Faheem Khan, Muhammad Khan, Taeg Whangbo, Critical challenges to adopt DevOps culture in software organizations: A systematic review, IEEE Access 10 (2022) 1, http://dx.doi.org/10.1109/ACCESS.2022.3145970.

[32] Ricardo Colomo-Palacios, Eduardo Fernandes, Pedro Soto-Acosta, Xabier Larrucea, A case analysis of enabling continuous software deployment through knowledge management, Int. J. Inf. Manage. (ISSN: 0268-4012) 40 (2018) 186–189, http://dx.doi.org/10.1016/j.ijinfomgt.2017.11.005, URL https://www.sciencedirect.com/science/article/pii/S0268401217308782.

[33] Rubén Grande, Aurora Vizcaino, Félix O. Garcia, Is it worth adopting DevOps practices in global software engineering? Possible challenges and benefits, Comput. Stand. Interfaces (ISSN: 0920-5489) 87 (2024) 103767, http://dx.doi.org/10.1016/j.csi.2023.103767, URL https://www.sciencedirect.com/science/article/pii/S092054892300048X.

[34] Hitesh Kumar Sharma, Anuj Kumar, Sangeeta Pant, Mangey Ram, 1 DevOps: An introduction, in: DevOps: A Journey from Microservice To Cloud Based Containerization, River Publishers, 2023, pp. 1–16, Conference Name: DevOps: A Journey from Microservice to Cloud Based Containerization, URL https://ieeexplore.ieee.org/abstract/document/10243527.

[35] Ram Shankar Uraon, Rashmi Bharati, Kritika Sahu, Anshu Chauhan, Agile work practices and team creativity: the mediating role of team efficacy, J. Organiz. Effect. People Perform. (ISSN: 2051-6614) ahead-of-print (ahead-of-print) (2023) http://dx.doi.org/10.1108/JOEPP-04-2023-0115.

[36] Chaitanya Arun Sathe, Chetan Panse, Analyzing the impact of agile mindset adoption on software development teams productivity during COVID-19, J. Adv. Manag. Res. (ISSN: 0972-7981) 20 (1) (2022) 96–115, http://dx.doi.org/10.1108/JAMR-05-2022-0088, Publisher: Emerald Publishing Limited, URL https://doi.org/10.1108/JAMR-05-2022-0088.

[37] Jan vom Brocke, Sonia Lippe, Managing collaborative research projects: A synthesis of project management literature and directives for future research, Int. J. Proj. Manage. (ISSN: 0263-7863) 33 (5) (2015) 1022–1039, http://dx.doi.org/10.1016/j.ijproman.2015.02.001, URL https://www.sciencedirect.com/science/article/pii/S0263786315000344.

[38] Filippo Lanubile, Christof Ebert, Rafael Prikladnicki, Aurora Vizcaíno, Collaboration tools for global software engineering, IEEE Softw. (ISSN: 1937-4194) 27 (2) (2010) 52–55, http://dx.doi.org/10.1109/MS.2010.39, Conference Name: IEEE Software, URL https://ieeexplore.ieee.org/document/5420797.

[39] Michal Hron, Nikolaus Obwegeser, Why and how is scrum being adapted in practice: A systematic review, J. Syst. Softw. (ISSN: 0164-1212) 183 (2022) 111110, http://dx.doi.org/10.1016/j.jss.2021.111110, URL https://www.sciencedirect.com/science/article/pii/S0164121221002077.

[40] Maja Due Kadenic, Konstantinos Koumaditis, Louis Junker-Jensen, Mastering scrum with a focus on team maturity and key components of scrum, Inf. Softw. Technol. (ISSN: 0950-5849) 153 (2023) 107079, http://dx.doi.org/10.1016/j.infsof.2022.107079, URL https://www.sciencedirect.com/science/article/pii/S0950584922001884.

[41] Lucas Gren, Vard Antinyan, On the relation between unit testing and code quality, in: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2017, pp. 52–56, http://dx.doi.org/10.1109/SEAA.2017.36.

[42] J.R. Horgan, S. London, M.R. Lyu, Achieving software quality with testing coverage measures, Computer (ISSN: 1558-0814) 27 (9) (1994) 60–69, http://dx.doi.org/10.1109/2.312032, Conference Name: Computer.

[43] Gabriele Bavota, Abdallah Qusef, Rocco Oliveto, Andrea De Lucia, David Binkley, An empirical analysis of the distribution of unit test smells and their impact on software maintenance, in: 2012 28th IEEE International Conference on Software Maintenance, ICSM, 2012, pp. 56–65, http://dx.doi.org/10.1109/ICSM.2012.6405253, ISSN: 1063-6773.

[44] Sina Shamshiri, José Miguel Rojas, Juan Pablo Galeotti, Neil Walkinshaw, Gordon Fraser, How do automatically generated unit tests influence software maintenance? in: 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), 2018, pp. 250–261, http://dx.doi.org/10.1109/ICST.2018.00033.

[45] Mark Grechanik, Chen Fu, Qing Xie, Automatically finding performance problems with feedback-directed learning software testing, in: 2012 34th International Conference on Software Engineering, ICSE, 2012, pp. 156–166, http://dx.doi.org/10.1109/ICSE.2012.6227197, ISSN: 1558-1225.

[46] Sacha Reis, Andreas Metzger, Klaus Pohl, A reuse technique for performance testing of software product lines, 2023.

[47] Emelie Engstrom, Per Runeson, Mats Skoglund, A systematic review on regression test selection techniques, Inf. Softw. Technol. (ISSN: 0950-5849) 52 (1) (2010) 14–30, http://dx.doi.org/10.1016/j.infsof.2009.07.001, URL https://www.sciencedirect.com/science/article/pii/S0950584909001219.

[48] Toni Mattis, Patrick Rein, Falco Dursch, Robert Hirschfeld, Rtptorrent: An open-source dataset for evaluating regression test prioritization, in: Proceedings of the 17th International Conference on Mining Software Repositories, MSR '20, Association for Computing Machinery, New York, NY, USA, ISBN: 978-1-4503-7517-7, 2020, pp. 385–396, http://dx.doi.org/10.1145/3379597.3387458, URL https://dl.acm.org/doi/10.1145/3379597.3387458.

[49] Sarika Chaudhary, Regression testing in agile: Concepts, strategies and challenges, Int. J. Res. Advent Technol. 7 (2019) 418–421, http://dx.doi.org/10.32622/ijrat.752019218.

[50] Gaoussou Abdel Kader Doukoure, Ernest Mnkandla, Facilitating the management of agile and devops activities: Implementation of a data consolidator, in: 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems, IcABCD, 2018, pp. 1–6, http://dx.doi.org/10.1109/ICABCD.2018.8465451, URL https://ieeexplore.ieee.org/abstract/document/8465451.

[51] Anitha Devi Nagarajan, Sietse J. Overbeek, A DevOps implementation framework for large agile-based financial organizations, in: Hervé Panetto, Christophe Debruyne, Henderik A. Proper, Claudio Agostino Ardagna, Dumitru Roman, Robert Meersman (Eds.), On the Move To Meaningful Internet Systems. OTM 2018 Conferences, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, ISBN: 978-3-030-02610-3, 2018, pp. 172–188, http://dx.doi.org/10.1007/978-3-030-02610-3_10.

[52] R. Alt, G. Auth, C. Kogler, Transformation of consulting for software-defined businesses: Lessons from a devops case study in a german it company, Contrib. Manag. Sci. (ISSN: 1431-1941) (2019) 385–403, http://dx.doi.org/10.1007/978-3-319-95999-3_19.

[53] S.M. Faizi, S. Rahman, K. Hopkins, Implementing large enterprise resource planning systems with agile methods, ISBN: 978-1-72816-309-3, 2019, http://dx.doi.org/10.1109/ICIET48527.2019.9290587.

[54] Nasreen Azad, Sami Hyrynsalmi, DevOps critical success factors — A systematic literature review, Inf. Softw. Technol. (ISSN: 0950-5849) 157 (2023) 107150, http://dx.doi.org/10.1016/j.infsof.2023.107150, URL https://www.sciencedirect.com/science/article/pii/S0950584923000046.

[55] Poonam Narang, Pooja Mittal, Vritti Narang, Automated continuous deployment of software projects with jenkins through DevOps-based hybrid model, 2023, http://dx.doi.org/10.21203/rs.3.rs-3205341/v1.

[56] Kalle Rindell, Jukka Ruohonen, Johannes Holvitie, Sami Hyrynsalmi, Ville Leppanen, Security in agile software development: A practitioner survey, Inf. Softw. Technol. (ISSN: 0950-5849) 131 (2021) 106488, http://dx.doi.org/10.1016/j.infsof.2020.106488, URL https://www.sciencedirect.com/science/article/pii/S0950584920302305.

[57] Alok Mishra, Ziadoon Otaiwi, DevOps and software quality: A systematic mapping, Comp. Sci. Rev. (ISSN: 1574-0137) 38 (2020) 100308, http://dx.doi.org/10.1016/j.cosrev.2020.100308, URL https://www.sciencedirect.com/science/article/pii/S1574013720304081.

[58] Dawei Yang, Daojiang Wang, Dongming Yang, Qiwen Dong, Ye Wang, Huan Zhou, Daocheng Hong, DevOps in practice for education management information system at ECNU, Procedia Comput. Sci. (ISSN: 1877-0509) 176 (2020) 1382–1391, http://dx.doi.org/10.1016/j.procs.2020.09.148, URL https://www.sciencedirect.com/science/article/pii/S1877050920320482.

[59] Claude Baron, Vincent Louis, Framework and tooling proposals for agile certification of safety-critical embedded software in avionic systems, Comput. Ind. (ISSN: 0166-3615) 148 (2023) 103887, http://dx.doi.org/10.1016/j.compind.2023.103887, URL https://www.sciencedirect.com/science/article/pii/S0166361523000374.

[60] V. Moysiadis, K. Tsakos, P. Sarigiannidis, E.G.M. Petrakis, A.D. Boursianis, S.K. Goudos, A cloud computing web-based application for smart farming based on microservices architecture, ISBN: 978-1-66546-717-9, 2022, http://dx.doi.org/10.1109/MOCAST54814.2022.9837727.

[61] Welder Pinheiro Luz, Gustavo Pinto, Rodrigo Bonifacio, Adopting DevOps in the real world: A theory, a model, and a case study, J. Syst. Softw. (ISSN: 0164-1212) 157 (2019) 110384, http://dx.doi.org/10.1016/j.jss.2019.07.083, URL https://www.sciencedirect.com/science/article/pii/S0164121219301517.

[62] Pooja Batra, Aman Jatain, Measurement based performance evaluation of DevOps, in: 2020 International Conference on Computational Performance Evaluation, ComPE, 2020, pp. 757–760, http://dx.doi.org/10.1109/ComPE49325.2020.9200149, URL https://ieeexplore.ieee.org/document/9200149.

[63] Ruth W. Macarthy, Julian M. Bass, An empirical taxonomy of DevOps in practice, in: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, 2020, pp. 221–228, http://dx.doi.org/10.1109/SEAA51224.2020.00046, URL https://ieeexplore.ieee.org/document/9226359.

[64] Anna Wiedemann, IT governance mechanisms for DevOps teams - how incumbent companies achieve competitive advantages, 2018, http://dx.doi.org/10.24251/HICSS.2018.617.

[65] Daniel Teixeira, Ruben Pereira, Telmo Henriques, Miguel Silva, João Faustino, A systematic literature review on DevOps capabilities and areas, Int. J. Hum. Capital Inf. Technol. Professionals 11 (2020) 1–22, http://dx.doi.org/10.4018/IJHCITP.2020040101.

[66] Theo Theunissen, Uwe van Heesch, Paris Avgeriou, A mapping study on documentation in continuous software development, Inf. Softw. Technol. (ISSN: 0950-5849) 142 (2022) 106733, http://dx.doi.org/10.1016/j.infsof.2021.106733, URL https://www.sciencedirect.com/science/article/pii/S095058492100183X.

[67] Muhammad Waseem, Peng Liang, Mojtaba Shahin, A systematic mapping study on microservices architecture in DevOps, J. Syst. Softw. (ISSN: 0164-1212) 170 (2020) 110798, http://dx.doi.org/10.1016/j.jss.2020.110798, URL https://www.sciencedirect.com/science/article/pii/S0164121220302053.

[68] Romina Eramo, Michele Tucci, Daniele Di Pompeo, Vittorio Cortellessa, Antinisca Di Marco, Davide Taibi, Architectural support for software performance in continuous software engineering: A systematic mapping study, J. Syst. Softw. (ISSN: 0164-1212) 207 (2024) 111833, http://dx.doi.org/

10.1016/j.jss.2023.111833, URL https://www.sciencedirect.com/science/article/pii/S0164121223002285.

[69] Mohammad Javad Kargar, Alireza Hanifizade, Automation of regression test in microservice architecture, in: 2018 4th International Conference on Web Research, ICWR, 2018, pp. 133–137, http://dx.doi.org/10.1109/ICWR.2018.8387249, URL https://ieeexplore.ieee.org/abstract/document/8387249.

[70] Engineer Bainomugisha, Alex Mwotil, Crane cloud: A resilient multi-cloud service abstraction layer for resource-constrained settings, Develop. Eng. (ISSN: 2352-7285) 7 (2022) 100102, http://dx.doi.org/10.1016/j.deveng.2022.100102, URL https://www.sciencedirect.com/science/article/pii/S2352728522000112.

[71] Rakesh Kumar, Rinkaj Goyal, Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC), Comput. Secur. (ISSN: 0167-4048) 97 (2020) 101967, http://dx.doi.org/10.1016/j.cose.2020.101967, URL https://www.sciencedirect.com/science/article/pii/S0167404820302406.

[72] Petri Ahokangas, Oxana Gisca, Marja Matinmikko-Blue, Seppo Yrjola, Jillian Gordon, Toward an integrated framework for developing European 6G innovation, Telecommun. Policy (ISSN: 0308-5961) 47 (9) (2023) 102641, http://dx.doi.org/10.1016/j.telpol.2023.102641, URL https://www.sciencedirect.com/science/article/pii/S0308596123001520.