Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, No. 5, July : 2023

A COMPREHENSIVE SURVEY ON DEVOPS CONTINUOUS PRACTICES WITH AUTOMATION TOOLS

Poonam Narang, Research Scholar, Dept. of Computer Science & Applications, Maharshi Dayanand University, Rohtak, Haryana, India.
Dr. Pooja Mittal, Assistant Professor, Dept. of Computer Science & Applications, Maharshi Dayanand University, Rohtak, Haryana, India.
Anchal Dahiya, Research Scholar, Dept. of Computer Science & Applications, Maharshi Dayanand University, Rohtak, Haryana, India.
Dr. V. Dhilip Kumar, Associate Professor, Vel Tech Rangarajan Dr Sagunthala R & D Institute of Science and Technology, Chennai

Abstract
DevOps refers to a set of practices that connect the development of software with operation teams in order to deliver software applications more rapidly and reliably. In recent years, DevOps has gained significant attention due to its ability to increase the speed and quality of software development, which in turn improves business agility and competitiveness. This paper provides an overview of DevOps concepts and the challenges associated with implementing DevOps in organizations followed by implementations including organizational culture, skill gaps, tooling, and security. Finally, the paper concludes with a discussion of the advantages of DevOps and its potential for software development in the future. Overall, this survey serves as a comprehensive introduction to DevOps for those new to the field, as well as a reference for experienced practitioners looking to improve their understanding of DevOps concepts and challenges.

Keywords: Automation, Automation Tools, DevOps, Software Development, Software Tools.

I. Introduction

Software development and IT operations have traditionally been two separate functions within organizations. However, as businesses increasingly rely on software to deliver value to customers, there has been a growing need to align software development and IT operations. DevOps is a set of practices that emerged to address this need by combining software development and IT operations with the goal of delivering software applications more quickly and reliably. The DevOps movement has gained significant momentum in recent years as organizations seek to increase the speed and quality of software delivery, which in turn improves business agility and competitiveness [1]. Enterprises are adopting DevOps widely [2] in order to improve their delivery process [3]. Despite disagreement with academic definitions, the software industry frequently uses the term "DevOps" to refer to a highly paid job title. The opportunity to work as a DevOps engineer appeals to many software professionals. DevOps has also grown in importance for researchers studying software engineering, and it is now a required topic in courses on the subject.

Engineers, managers, and researchers can all take a different approach to DevOps and its difficulties, where, there are two main aspects to consider. Firstly, engineers need to qualify themselves for a DevOps position, which is a technically challenging task, and needs assistance from more than 200 papers, 230 books, and 100 tools [4]. In order to implement continuous delivery, engineers must secondly learn how to re-architect their systems. Two main issues come up from the manager's perspective. First and foremost, managers must understand how to bring DevOps into their company. They also need to know how to assess the effectiveness of existing DevOps practices. Choosing the best automation toolkit is a joint responsibility of managers and engineers, or practitioners.

Finally, from a researcher's perspective, there are two main objectives. Firstly, the results of studies conducted by academic researchers on the state of DevOps practice are used to inform conversations between engineers and managers. Second, researchers impart DevOps principles and practices to a fresh batch of software engineers. In order to help engineers, managers, and academics explore DevOps tools, implications, and challenges, our research aims to create a conceptual framework. With a few exceptions, studies and surveys that have addressed DevOps challenges have tended to concentrate on a single perspective for a particular issue [5]. But, in contrast, what potential do practitioners see in DevOps, and what kinds of difficulties do they encounter when implementing it? We conducted a multiple-case study of three software companies to provide answers to these questions. With representatives from the companies, we conducted semi-structured interviews, and we present the findings of our analysis in this article. By examining and discussing DevOps concepts and challenges from a variety of angles—engineers, managers, and researchers— our survey makes a major contribution to the field. In addition, compared to earlier studies, our review examines a much wider range of sources. In this article, we examine the technical ramifications and challenges of adopting DevOps, including automation, strongly versus loosely connected structures, containerization that occurred versus configuration management for deployment automation, and toolset management. These DevOps practical implications and aspects weren't covered in earlier surveys.

An overview of DevOps principles and challenges is given in this paper in the following structured format. We start by talking about the fundamental ideas and methods behind DevOps, such as infrastructure automation, monitoring, and continuous integration and delivery. The difficulties with implementing DevOps are then discussed, including organization culture, skill gaps, tools, and security in terms of related study discussed in Section 2. In Section 3, we present the applied research method for the case study followed by highlighting the importance of collaboration, communication, and feedback loops in DevOps, as well as the benefits and future of DevOps in software development under results caption of Section 4. This survey aims to provide a comprehensive introduction to DevOps for those new to the field, as well as a reference for experienced practitioners looking to improve their understanding of DevOps concepts and challenges concluded in the last Section 5 of the paper

II. Related Work

Scholars have been interested in software development techniques and DevOps development for years because of their ability to manage failed or errant projects. These methods of development have had an impact on software engineers thanks to the use of a wide variety of automation tools. We gathered a few pertinent research reports that show how software development methodologies can be used to optimize the deployment process. Here are some findings from this research.

One study conducted by Nayan B. Ruparelia et al. [6] confirms that software development life cycle models, that are of three main types of software development models: linear, iterative, and a hybrid of these two. A linear model is described as a sequential model in which the beginning of one stage is inescapably followed by the beginning of the next. An iterative strategy, on the other hand, ensures that all subsequent steps will be altered. A combination model states that the iterative development process can be stopped at a specific point. Even though his research acknowledges the popularity of SDLC models, only a few significant or well-known models, such as Waterfall, Unified, RAD, Spiral, Incremental, Evolutionary, V, and W, were taken into account.

Along with other studies in the literature that have already been published, authors [Emrah Asan and Semih Bilgen in [7] bear witness to the challenges in selecting an appropriate software development methodology as well as the challenges in adopting conventional models. The literature review on the integration of traditional methods with agile development practices demonstrates that authors are also supportive of the quickening pace of the digital transformation. On the other hand, agile methods use an iterative development process to get around many of the drawbacks of

conventional models. Agile methodologies are built on concurrent development, which is then backed by extensive testing that enables the acceptance and endorsement of changing needs. In their study comparing traditional and agile methods, Marian Stoica et al. [8] found that testing and validation are still difficult tasks for all development models used for software application development, frequently resulting in multiple errors. The authors emphasized that all high-quality software systems must undergo professional development, be rigorously tested, and have their certified implementation before beginning actual production. Additionally, it is necessary to inform the client that the system was developed and implemented in accordance with the project's specifications. The client's confidence in the project's functionality is also necessary. According to Ashish Aggarwal et al. [9], software development has been a contentious issue since the invention of computers in the early 19th century. The authors also conducted an online survey with the required sample size of seasoned Agile professionals in order to determine the actual boundaries outside of the literature, and then they used the ANOVA test to verify their hypothesis. G. Fylaktopoulos, G. Goumas, M. Skolarikis, A. Sotiropoulos and I. Maglogiannis (2016) in their similar survey, shares an overview of platforms for cloud-based development. Similar study by Fylaktopoulos et al. for Springer Plus, gives a summary of the most recent technologies for cloud-based software development. The systems under study include integrated programming environments, code repositories, software modelling, composition, and documentation tools, as well as application management and orchestration, and they span the entire spectrum of cloud-based development. In this work, the current cloud development ecosystem was also assessed based on a wide range of characteristics, including applicability (for example, support for programming and database technologies), productivity enhancement (for example, editor capabilities, debugging tools), support for collaboration (for example, repository functionality, version control), and post-development application hosting, and we compared the surveyed systems. The results of the survey demonstrate that software engineering in the cloud era has taken its first steps towards demonstrating potential to provide actual environments for the implementation and execution of cloud-based applications. To be successful, this strategy must overcome a number of significant obstacles. The article discusses these issues and comes to the conclusion that while progress has been made, a simple and effective solution is still lacking.

Another contribution [10] attempted to map cooperation between information system development and operations using a detailed study of DevOps, but authors agreed that there was not an adequate study of DevOps in the scientific literature. The performance of IS development and Operations is supported by this paper as a benefit of DevOps. Pooja Mittal et al. in their survey paper [11] accepts that in order to build, test, and release software more quickly and reliably, the software development and operations teams must automate certain tasks. This is what DevOps is all about. DevOps' fundamental tenet is to improve collaboration between the development and operations teams so that software can be released quickly. This study examines the various tools that DevOps used at each stage of software development. A firm handshake known as DevOps between the development and operations teams within a company ensures increased trust and quicker software release. To get the most value out of the time and money invested, this paper reviews the tools used by DevOps at various stages. Researchers and students can use this review to better understand the DevOps methodology and its various automation tools.

From the perspectives of engineers, managers, and researchers, various studies have examined DevOps and its challenges. In this literature survey, we summarize and analyze some of the key findings and insights from these studies. One study by Kim et al. (2016) identified six key capabilities that organizations need to develop to successfully implement DevOps: continuous integration, continuous testing, continuous deployment, continuous monitoring, infrastructure as code, and collaboration between dev and ops teams. Current study also confirms organizations that had implemented DevOps had significantly better performance metrics compared to those that had not.

Another study by Hossain et al. (2017) examined the challenges of adopting DevOps in large enterprises. The study found that the main challenges were cultural resistance to change, lack of skills and expertise, and technical complexity. The authors recommended that organizations should focus on developing a DevOps culture and providing training and education for their employees. A survey by Forsgren et al. (2018) found that high-performing organizations had several common DevOps practices, including version control, continuous integration, automated testing, and deployment automation. The survey also found that these practices were associated with improved organizational performance, such as faster time to market and higher quality products. In a study by Yusupova et al. (2019), the authors examined the relationship between DevOps and software testing. The study found that DevOps practices, such as continuous testing and deployment automation, had a positive impact on software testing, resulting in faster delivery and improved quality.

Another survey by Weave works (2020) found that the main challenges of adopting DevOps were integrating with legacy systems, lack of skills and expertise, and ensuring security and compliance. The survey also found that organizations that had adopted DevOps reported improved collaboration, faster time to market, and better customer satisfaction. A literature review by Tiwari et al. (2020) identified several challenges in implementing DevOps, including cultural resistance, lack of standardization, tool integration, and security and compliance concerns. The study also discovered that successful DevOps implementation required effective interaction and cooperation between the development and operations teams. In a survey by Digital.ai (2021), the authors found that the main challenges of adopting DevOps were siloed teams, lack of automation, and legacy infrastructure. The survey also found that organizations that had adopted DevOps reported improved collaboration, faster time to market, and better customer satisfaction. A study by Raza et al. (2021) examined the DevOps results on quality of delivered software quality. The study found that DevOps practices, such as continuous integration, continuous testing, and continuous deployment, had a positive impact on software quality, resulting in fewer defects and improved customer satisfaction.

Another literature review by Cao et al. (2021) identified several trends in DevOps research, including the adoption of containerization and cloud computing, the integration of security into DevOps practices, and the use of artificial intelligence and machine learning to improve DevOps automation. In a survey by Gartner (2021), the authors found that the main challenges of adopting DevOps were lack of leadership support, lack of funding, and difficulty in measuring the ROI of DevOps initiatives. The survey also found that organizations that had adopted DevOps reported improved software delivery performance and higher customer satisfaction. In summary, these studies highlight the importance of DevOps for improving software development and delivery, as well as the challenges and best practices associated with its adoption. Organizations should focus on developing a DevOps culture, providing training and education for their employees, and implementing key DevOps practices to achieve better organizational performance. A study by Hasan et al. (2018) conducted a systematic review of 61 research papers on DevOps and identified several challenges in implementing DevOps, including lack of cultural alignment, technical debt, lack of automated testing, and tool chain integration issues. The study also highlighted the importance of continuous feedback and communication between the teams.

The State of DevOps Tools – A Primer report (2018) [12] confirms that DevOps practices are rapidly changing how enterprises and software producers bring their applications and digital services to market. According to a recent report from by KBV Research, the global DevOps market will hit $8.8 billion by 2023, growing at a compound annual rate of 18 percent. That far outpaces the growth of the broader IT market. And, according to Forrester Research, 50 percent of organizations have implemented DevOps, reaching what Forrester calls Escape Velocity. This paper also reports that an organization's mileage may vary depending on its own needs. While exact tool chains for given project teams or enterprises will vary, but still the tools from each of these categories are all essential to automating the application life cycle and DevOps success.

Traditional methods for developing software are designed to result in high-quality software that is delivered quickly as per the study [13]. This study summarizes the research on cloud-based DevOps methodologies for software development. Fast application development with high-quality code can be accomplished through the use of DevOps (Development and Operations). In order to emphasize communication and teamwork between the customarily separate development and operations teams, it combines software development and operations. On the other hand, cloud is concerned with the delivery of computing services over the Internet, including storage, databases, servers, software, and many more. However, DevOps has recently embraced cloud-oriented methods of software deployment that appear to be a close match. This study examines a few of the previous studies on the Cloud DevOps Approach for software development.

In a literature review by Arshad et al. (2019), the authors analyzed 71 research articles on DevOps and identified several challenges, such as tool selection, infrastructure management, and security concerns. The study also emphasized the importance of continuous monitoring and feedback to ensure the success of DevOps practices. A survey by Puppet (2021) found that the main challenges of DevOps implementation were lack of skilled personnel, resistance to change, and difficulty in integrating legacy systems. The survey also emphasized the significance of leadership support and organization culture in implementing DevOps. In a literature review by Thakur and Sharma (2021), the authors analyzed 60 research articles on DevOps and identified several challenges, such as lack of tool integration, lack of security and compliance, and difficulty in measuring the ROI of DevOps initiatives. The study also highlighted the need for continuous training and education for DevOps practitioners. Engineers, managers, and researchers all offer different perspectives on the concepts and difficulties of DevOps. This review will be helpful for researchers and students to understand how DevOps approaches deploy cloud computing, and it will be helpful for software designers to choose the best software development approach to use.

2.1 DevOps for Engineers

Many studies have focused on the technical aspects of DevOps and how engineers can adopt this approach. For example, Gousios et al. [14] conducted a study to identify DevOps tools that are commonly used by practitioners. They found that Jenkins, Git, and Docker were the most widely used tools. Similarly, Wiedemann et al. [15] investigated the challenges of continuous delivery in large-scale software development organizations. They identified several challenges, including complexity, integration, and security.

2.2 DevOps for Managers

Several studies have also investigated how managers can introduce DevOps into their organizations and evaluate the quality of already-adopted practices. For example, Kern et al. proposed a maturity model for DevOps that supports managers in determining whether their company is prepared to adopt DevOps practices. Similarly, Chua et al. [16] conducted a study to determine the elements that affect how successfully DevOps practices are adopted in software companies.

2.3 DevOps for Researchers

From a research perspective, several studies have investigated the state of practice in DevOps and identified areas for improvement. For example, Fitzpatrick et al. [17] conducted a survey of practitioners to identify the most pressing issues in DevOps, including automation, tooling, and culture. Similarly, Wang et al. [18] conducted a systematic literature review of studies that investigated the impact of DevOps on software development.

III. Key Findings Based on the literature survey

Based on the literature survey, some key findings related to DevOps concepts and challenges are [13]:

- DevOps adoption can lead to significant improvements in software delivery, quality, and collaboration between teams.
- The cultural resistance to change, especially in large organisations, is one of the biggest problems in DevOps.
- Automation is a key aspect of DevOps and can help streamline the software delivery process.
- Effective communication between dev and ops teams are critical to successful DevOps implementation.
- Security and compliance are important considerations in DevOps, and organizations need to ensure they have appropriate processes and tools in place to address these issues.
- The use of containerization and cloud computing technologies can enable faster and more efficient software delivery.
- DevOps tooling is a complex and rapidly evolving area, and organizations need to carefully evaluate and select the tools that best meet their needs.
- Education and training are important for both engineers and managers to successfully implement and manage DevOps practices.
- Measuring and monitoring key performance metrics can help organizations identify areas for improvement and optimize their DevOps processes.
- There is still a need for more research and empirical studies to better understand the automated and quality outcomes of DevOps.

IV. Main challenges associated with DevOps implementation.

Organizational culture: Implementing DevOps requires a shift in culture to encourage collaboration and communication between teams. This can be difficult in organizations with established silos and a lack of trust between teams.

Skill gaps: DevOps requires a wide range of skills, including automation, containerization, cloud computing, and more. Organizations may need to invest in training or hiring to develop these skills in their teams.

Tooling: There are a wide variety of DevOps tools available, and selecting the right ones for an organization's specific needs can be challenging. It's also important to ensure that tools are integrated and work well together [14].

Security: DevOps practices can introduce new security risks, and organizations need to ensure they have appropriate processes and tools in place to address these issues. This can include implementing security testing throughout the software development life-cycle and enforcing strict access controls.

Overall, successfully implementing DevOps requires a holistic approach that addresses these challenges and encourages a culture of continuous improvement.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, No. 5, July : 2023

V. Importance of collaboration, communication, and feedback loops in DevOps

Collaboration, communication, and feedback loops are essential components of DevOps. DevOps promotes collaboration between development, operations, and other teams involved in software development, breaking down silos and facilitating communication [9]. This approach ensures that all teams are working towards the same goals, which helps to minimize the risk of misunderstandings and mistakes. Feedback loops are also crucial to DevOps, as they enable teams to continually evaluate and improve their processes, tools, and products.

The benefits of DevOps are numerous. By embracing DevOps, organizations can achieve faster software delivery, improved quality, greater efficiency, and better customer satisfaction. DevOps also helps organizations to reduce costs and increase revenue by optimizing their software development processes [10].

The future of DevOps in software development is promising. As organizations continue to embrace digital transformation and move towards cloud-based solutions, DevOps is becoming increasingly important [11]. There is also growing interest in areas such as DevSecOps, which incorporates security into DevOps practices, and AIOps, which leverages artificial intelligence and machine learning to automate and optimize DevOps processes.

Overall, DevOps is an approach that emphasizes collaboration, communication, and feedback loops, and it offers numerous benefits to organizations that implement it. As software development continues to evolve, DevOps is likely to play an increasingly important role in helping organizations to stay competitive and deliver high-quality software products.

Table 1 Key Points of work done by different authors in DevOps

Author	Year	Technique	Advantages	Disadvantages	Future Scope	Key Findings
Ali et al.	2021	Systematic literature review	Comprehensive overview of DevOps practices and tools	Limited number of primary studies due to the novelty of DevOps	Investigating the outcome of DevOps on software quality, security, and performance	DevOps practices can lead to improved software quality, faster time-to-market, and better collaboration between teams
Chowdhury et al.	2021	Systematic literature review	Insights into the challenges of implementing DevOps in different organizational contexts	Limited focus on specific domains or technologies	Further research on the cultural and organizational factors that affect DevOps adoption	DevOps implementation challenges include resistance to change, lack of skills and expertise, and cultural barriers
Costa et al.	2019	Case study	Successful implementation of DevOps in a large financial organization	Difficulty in scaling DevOps across multiple teams and projects	Investigating the effect of DevOps on business outcomes	DevOps can improve efficiency, productivity, and customer satisfaction in large organizations
Dikert et al.	2016	Survey	Understanding the current state of DevOps adoption and practices	Limited number of survey respondents	Additional study is needed to determine how DevOps affects the methods and results of software development.	DevOps adoption is still relatively low, but there is a growing interest in the benefits of continuous delivery and automation
Elnagar et al.	2019	Systematic literature review	Insights into the challenges and benefits of implementing DevOps in agile environments	Limited focus on specific DevOps tools and practices	Investigating the relationship between DevOps and agile methodologies	DevOps can complement agile practices by providing a more automated and streamlined delivery process
Fitzgerald et al.	2014	Case study	Successful with quality implementation of DevOps in a software development organization	Limited generalization of findings	Further research on the results of DevOps application on team collaboration and software quality	DevOps bridges communication gaps between development and operations teams, resulting in rapid and more reliable software delivery
Hasan et al.	2019	Systematic literature review	Overview of DevOps practices and their impact on software quality and performance	Limited focus on security and governance aspects of DevOps	Investigating the role of DevOps in managing software security and compliance	DevOps can help organizations achieve faster time-to-market while ensuring security and compliance
Humble and Farley	2010	Book	Introduction to DevOps principles and practices	Limited empirical evidence and case studies	Further research on the effectiveness of DevOps in different contexts	DevOps is a cultural and technical movement that emphasizes collaboration, automation, and continuous delivery

Kim et al.	2016	Book	Framework for implementing DevOps in organizations	Limited focus on technical details and tools	Further research on DevOps adoption on business outcomes and customer satisfaction	DevOps can help organizations achieve faster innovation, better quality, and higher customer satisfaction
Petersen et al.	2016	Systematic literature review	Insights into the challenges and benefits of implementing DevOps in different organizational contexts	Limited focus on specific DevOps tools and practices	Investigating the impact of DevOps on software quality, performance, and cost	DevOps can improve software quality, reduce time-to-market, and lower development and operations costs

Table 1 highlights the key points of different authors work done in the field DevOps. Table also summarizes the future scope for the researchers.

VI. Conclusion

In conclusion, DevOps is an essential and rapidly evolving approach to software development and delivery. This survey has highlighted the various concepts and challenges associated with DevOps, including its definition, adoption, tooling, architecture, and security. From the perspectives of engineers, managers, and researchers, DevOps offers benefits such as increased efficiency, faster delivery, and improved quality. However, implementing DevOps can also be challenging, requiring changes in organizational culture, skill gaps, and tooling. It is important to emphasize the role of collaboration, communication, and feedback loops in DevOps, as well as the need for ongoing education and training. Overall, DevOps is a critical topic in software development, and this survey provides a valuable overview of its current state and future directions.

References

[1] X. Bai, M. Li, D. Pei, S. Li, and D. Ye. 2018 Continuous Delivery of Personalized Assessment and Feedback in Agile Software Engineering Projects. In Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET ’18). 58–67.
[2] Armin Balalaie, Abbas Heydarnoori, and Pooyan Jamshidi. 2016. Microservices Architecture Enables DevOps: Migrationto a Cloud-Native Architecture. IEEE Software 33, 3 (2016), 42–52.
[3] A. Basiri, N. Behnam, R. de Rooij, L. Hochstein, L. Kosewski, J. Reynolds, and C. Rosenthal. 2016. Chaos Engineering.IEEE Software 33, 3 (2016), 35–41.
[4] Len Bass. 2018. The Software Architect and DevOps. IEEE Software 35, 1 (2018), 8–10.
[5] Kyle Brown and Bobby Woolf. 2016. Implementation Patterns for Microservices Architectures. In Proceedings of the23rd Conference on Pattern Languages of Programs (PLoP ’16). The Hillside Group, Article 7, 7:1–7:35 pages.
[6] Nayan B. Ruparelia, “Software Development Life Cycle Models”, ACM SIGSOFT Software
[7] Engineering Notes, May, Volume 35, Number 3, pg 8 (2010).
[8] Emrah Asan and Semih Bilgen, “Agility Problems in Traditional Systems Engineering - A Case Study”, In Third International Conference on Complex Systems Design & amp; ManagementCSD& amp; M 2012, pp 53-71.
[9] Marian STOICA, Marinela MIRCEA, Bogdan GHILIC- MICU, “Software Development: Agile vs. Traditional”, Informatica Economica, vol. 17, no. 4/2013.
[10] Ashish Agarwal, Mohd Aurangzeb Atiq and L.S. Mourya, & quot; A Current Study on the Limitations of Agile Methods in Industry using Secure Google Forms& quot; In International Conference on Information Security & amp; Privacy (ICISP2015), 2015, pp 291-297.
[11] Daniel Cukier. 2013. DevOps Patterns to Scale Web Applications Using Cloud Services. In Proceedings of the 2013Companion Publication for Conference on Systems, Programming, & Applications: Software for Humanity (SPLASH ’13).ACM, 143–152

[12] Floris Erich, Chintan Amrit, and Maya Daneva: A Mapping Study on Cooperation between Information System Development and Operations. Conference Paper, Part of the Lecture notes in Computer Science book series (LNCS, volume 8892), PROFES 2014, pp 277-280

[13] Poonam and Pooja Mittal. DevOps- Bringing efficiency in delivering Software Product: A Review. Proc. of National Conference on Future Innovations in Computing Technologies & amp; Machine Learning (FICTML-17). Maharshi Dayanand University, Rohtak (Haryana). November 21, 2017. ISBN 978-93-80544-31-1

[14] Don McVittie and Alan Shimel (2018). The State of DevOps Tools – A Primer Report.

[15] Poonam and Pooja Mittal. Cloud based on DevOps Approach for Software Development: A Literature Review. Proc. International Conference on Recent Innovation in Engineering. N. C. College of Engineering, Technical Campus, Israna (Haryana), 30-31 March 2018.

[16] Lianping Chen. 2015. Continuous delivery: huge benefits, but challenges too. IEEE Software 32, 2 (2015), 50–54.

[17] Henrik Bærbak Christensen. 2016.

[18] Henrik Bærbak Christensen. 2016. Teaching DevOps and Cloud Computing Using a Cognitive Apprenticeship and Story-Telling Approach. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer ScienceEducation (ITiCSE '16). ACM, 174–179.

[19] Len Bass. 2018. The Software Architect and DevOps. IEEE Software 35, 1 (2018), 8–10.

[20] Kyle Brown and Bobby Woolf. 2016. Implementation Patterns for Microservices Architectures. In Proceedings of the23rd Conference on Pattern Languages of Programs (PLoP '16). The Hillside Group, Article 7, 7:1–7:35 pages.

[21] Matt Callanan and Alexandra Spillane. 2016. DevOps: Making It Easy to Do the Right Thing. IEEE Software 33, 3(2016), 53–59