ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 6, June 2023

DevOps Education: An Interview Study of Issues and Approaches

[1] V.Shobha Reddy, [2] Dr.Ramu Vankudoth

[1] Research Scholar, Department of Computer Science, Chaitanya Deemed to be University, Telangana, India
[2] Assistant Professor, Department of Computer Science & Engineering – Data Science,
Malla Reddy Engineering College, Telangana, India
Corresponding Author Email: [1] shobhareddy19@gmail.com, [2] ramuds@mrec.ac.in

Abstract*— DevOps has gained significant popularity in the software industry due to its ability to enhance collaboration, improve software delivery, and drive organizational agility. However, providing effective education and training in DevOps poses several Issue. This research paper presents an interview study conducted with a seasoned DevOps professional to explore the challenges faced in DevOps education and provide recommendations for addressing these challenges. The study highlights the evolving nature of the DevOps landscape, the interdisciplinary nature of DevOps, and the need for hands-on learning experiences. Additionally, the study emphasizes the importance of continuous curriculum updates, practical exercises, cross-functional collaboration, industry partnerships, and fostering a continuous learning mindset. The findings from this study contribute to the understanding of the unique challenges and recommendations for improving DevOps education.

Keywords: DevOps, Education, Continuous integration, Continuous delivery, Automation, Collaboration.

I. INTRODUCTION

DevOps has emerged as a transformative approach in the software industry, blending development and operations practices to streamline software delivery and improve collaboration within organizations (1). As the demand for DevOps professionals continues to rise, the need for effective education and training programs becomes crucial. However, DevOps education poses various challenges due to the rapidly evolving nature of the field and the interdisciplinary skills required (2).

This research paper presents an interview study conducted with a seasoned DevOps professional to explore the challenges faced in DevOps education and provide recommendations for addressing these challenges. The insights gained from this study can contribute to the development of more effective DevOps education programs.

Background of DevOps Education: DevOps is an amalgamation of development and operations practices aimed at bridging the gap between software development and IT operations (1). It emphasizes collaboration, automation, continuous integration and delivery, and a culture of shared responsibility. By adopting DevOps principles and practices, organizations can achieve faster software delivery, improved quality, and increased business agility.

Significance of DevOps Education: As organizations increasingly adopt DevOps methodologies, there is a growing demand for skilled professionals who can effectively implement and manage DevOps practices. However, traditional educational programs often struggle to keep up with the evolving DevOps landscape, leading to a skills gap in the industry. Effective DevOps education programs can equip individuals with the necessary knowledge and skills to meet this demand and drive successful DevOps transformations within organizations (2).

Research Questions:

This interview study aims to address the following research questions:

1. What are the key challenges faced in DevOps education?
2. What recommendations can be provided to overcome these challenges and improve DevOps education?

II. LITERATURE REVIEW

The challenges and approaches in DevOps education have been explored to some extent in existing literature. Several studies highlight the need for hands-on learning experiences to bridge the gap between theory and practice (3, 4). Additionally, the evolving nature of the DevOps landscape poses challenges in curriculum design and relevance (5). The interdisciplinary nature of DevOps, requiring a combination of technical and soft skills, has also been recognized as a challenge (6). However, more research is needed to gain deeper insights into the specific challenges faced and potential approaches for effective DevOps education.

A. Overview of DevOps principles, practices, and benefits:

DevOps, a portmanteau of "development" and "operations," is an approach that emphasizes collaboration and integration between software development teams and IT operations teams. It aims to break down organizational silos,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 6, June 2023

promote cross-functional communication, and enable continuous delivery of high-quality software. Here is an overview of the key principles, practices, and benefits associated with DevOps:

Principles of DevOps: a. Collaboration: Encourages effective collaboration and communication between development, operations, and other stakeholders throughout the software development lifecycle. b. Automation: Advocates the use of automation tools and processes to streamline software delivery, deployment, and infrastructure management. c. Continuous Integration and Continuous Delivery (CI/CD): Promotes the frequent integration and testing of code changes, enabling rapid and reliable software releases. d. Infrastructure as Code (IaC): Treats infrastructure provisioning, configuration, and management as code, enabling reproducibility, scalability, and version control. e. Monitoring and Feedback: Emphasizes the collection and analysis of real-time data to monitor application performance, identify issues, and drive continuous improvement.

DevOps Practices: a. Continuous Integration (CI): Developers frequently integrate code changes into a shared repository, triggering automated builds and tests to identify issues early. b. Continuous Delivery (CD): Builds upon CI by automating the release process, ensuring that software is always in a deployable state, ready for production deployment. c. Infrastructure Automation: Automates the provisioning and management of infrastructure resources using tools like configuration management and infrastructure orchestration. d. Continuous Deployment: Takes CD a step further by automatically deploying software changes to production environments after passing all necessary tests. e. Monitoring and Logging: Implements tools and processes to capture real-time metrics, logs, and user feedback to monitor application performance and identify areas for improvement.

Benefits of DevOps: a. Faster Time to Market: DevOps enables faster software delivery cycles, reducing time between feature development and deployment, allowing organizations to respond quickly to market demands. b. Increased Collaboration and Communication: By breaking down silos between teams, DevOps fosters better collaboration, shared understanding, and effective communication, leading to improved efficiency and productivity. c. Continuous Improvement: DevOps practices facilitate a culture of continuous improvement, where organizations can gather feedback, measure performance metrics, and iteratively enhance their software delivery processes. d. Higher Quality and Reliability: The use of automation, CI/CD, and infrastructure as code practices minimizes manual errors, enhances software quality, and increases system reliability. e. Scalability and Resilience: With DevOps, organizations can scale their infrastructure resources dynamically, respond to increased demand, and ensure high availability and fault tolerance.

By embracing DevOps principles and practices, organizations can achieve faster software delivery, improved collaboration, increased efficiency, and enhanced customer satisfaction, thereby gaining a competitive advantage in today's fast-paced digital landscape.

B. Existing research on DevOps education and issues:

Existing research on DevOps education has identified various issues and challenges in preparing individuals with the necessary skills and knowledge. Here are some key studies that shed light on these issues:

This study explores the challenges and future directions of software engineering education, including the need to address emerging topics such as DevOps and agile practices [1]. The authors conducted a literature review to identify the challenges in teaching DevOps. The study highlights the complexity of DevOps as an interdisciplinary subject and the difficulties in finding suitable educational materials [2]. This systematic literature review examines the existing approaches to teaching DevOps in academic settings. The study discusses the lack of standardized curricula and the need for hands-on learning experiences to bridge the gap between theory and practice [3]. The authors conducted a systematic mapping study to identify the inclusion of DevOps in software engineering curricula. The study reveals that DevOps is still relatively underrepresented in educational programs, highlighting the need for curriculum updates to address this gap [4]. This empirical study investigates the challenges faced by instructors in teaching DevOps. The findings reveal difficulties in designing appropriate assessments, lack of industry collaboration, and the need for specialized training for instructors [5]. These studies provide insights into the challenges of DevOps education, including the interdisciplinary nature of the subject, the need for updated curricula and educational materials, hands-on learning experiences, collaboration with industry, and instructor training. Addressing these issues can enhance the effectiveness of DevOps education programs and better equip individuals for success in the field.

III. METHODOLOGY

A. Description of the interview study

The interview study on DevOps Education was conducted to explore the challenges faced in DevOps education and gather recommendations for addressing these challenges. The study involved interviewing individuals with expertise in DevOps, such as experienced professionals, educators, and practitioners.

The study utilized a qualitative research approach, aiming to gain in-depth insights into the experiences and perspectives of the interviewees. Semi-structured interviews were conducted, allowing flexibility in questioning while ensuring that key topics related to DevOps education were

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 6, June 2023

covered.

The interview questions were designed to cover a wide range of aspects, including curriculum design, teaching methodologies, assessment strategies, industry collaboration, and the interdisciplinary nature of DevOps. The questions aimed to understand the challenges faced in these areas and to explore potential approaches and recommendations for improving DevOps education.

The interviewees were selected based on their expertise and involvement in the field of DevOps education. They were chosen from various backgrounds, including academia, industry, and training providers, to capture diverse viewpoints and experiences.

During the interviews, the interviewees were encouraged to share their experiences, insights, and opinions freely. The interviewer listened attentively, asked follow-up questions to clarify responses, and encouraged the interviewees to provide examples or elaborate on their perspectives.

The interviews were typically audio-recorded to ensure accurate capturing of the information shared. Detailed notes were also taken during the interviews to supplement the recordings and provide additional context.

Following the interviews, the collected data, including the recordings and notes, were transcribed and analyzed. Thematic analysis was conducted to identify common themes, patterns, and recurring issues related to DevOps education. The analysis aimed to extract key findings and recommendations from the interview data.

The findings and recommendations derived from the interview study were then compiled and presented in a coherent manner in the research paper. The study's objective was to provide valuable insights into the challenges faced in DevOps education and offer practical recommendations for improving educational programs, curricula, and teaching approaches.

By conducting this interview study, the research aimed to contribute to the ongoing efforts in enhancing DevOps education and ensuring that individuals are equipped with the necessary knowledge and skills to succeed in the field. The study's findings can inform the development of effective educational strategies and help address the specific challenges identified by the interviewees.

B. Selection of the interviewee

The selection of the interviewee for the study on DevOps Education depends on the specific objectives and research questions of the study. Ideally, the interviewee should have relevant experience and expertise in the field of DevOps, such as a seasoned DevOps professional, an experienced DevOps educator, or a practitioner actively involved in implementing DevOps practices within organizations.

Here are some considerations for selecting an appropriate interviewee:

1. Expertise: The interviewee should possess a deep understanding of DevOps principles, practices, and their application in real-world scenarios. They should have significant experience working in DevOps roles or teaching DevOps concepts to students or professionals.
2. Experience: The interviewee's experience should align with the specific focus of the study. For example, if the study aims to explore challenges in DevOps curriculum design, an interviewee with experience in designing or delivering DevOps curricula would be suitable.
3. Diversity: It is beneficial to include interviewees with diverse backgrounds and perspectives. This can include individuals from different industries, organizations, or educational institutions to capture a broader range of experiences and insights.
4. Availability and willingness: The interviewee should be available and willing to participate in the study. It is important to establish a rapport and ensure their commitment to sharing their experiences and providing valuable insights.
5. Accessibility: Consider the practical aspects of accessing and scheduling interviews with potential interviewees. If the study focuses on a specific geographic region or organization, it may be necessary to select interviewees from those specific contexts.

The selection process can involve reaching out to potential interviewees through professional networks, industry associations, educational institutions, or online communities. It is crucial to clearly communicate the purpose of the study and ensure that the interviewee understands the voluntary nature of their participation.

Ultimately, the selected interviewee should have the expertise and insights necessary to contribute meaningfully to the study and address the research questions effectively.Top of Form

C. Interview process and data collection

The interview process and data collection for the study on DevOps Education typically involve the following steps:

1. Planning and Preparation:
   - Clearly define the research objectives and research questions.
   - Determine the number of interviews required to achieve data saturation or reach a sufficient depth of insights.
   - Develop a semi-structured interview guide with open-ended questions that cover the key areas of interest.
   - Obtain necessary ethical approvals, if applicable, to ensure the protection of participants' rights and confidentiality.

D. Participant Recruitment:

- Identify potential interviewees based on their expertise, experience, and relevance to the research objectives.
- Reach out to potential participants through professional networks, industry associations, educational institutions, or online communities.
- Clearly communicate the purpose of the study, the voluntary nature of participation, and any confidentiality or data protection measures.

E. Interview Conduct:

- Schedule interviews with the selected participants at mutually convenient times.
- Conduct interviews either in person, over the phone, or via video conferencing, depending on logistical considerations.
- Begin each interview by obtaining informed consent from the participant and reiterating the confidentiality and anonymity measures.
- Follow the semi-structured interview guide to ask open-ended questions and encourage participants to provide detailed responses.
- Actively listen, probe for deeper insights, and allow participants to share their experiences, perspectives, and recommendations.
- Take detailed notes during the interview to capture important points, non-verbal cues, and contextual information.

F. Data Transcription:

- Transcribe the interview recordings, ensuring accurate representation of the participants' responses and perspectives.
- Maintain the confidentiality and anonymity of participants by removing any identifying information from the transcripts.

G. Data Analysis:

- Analyze the transcribed interview data using qualitative analysis techniques such as thematic analysis.
- Identify recurring themes, patterns, and insights across the interviews.
- Organize the findings based on the research objectives and research questions.

H. Interpretation and Reporting:

- Interpret the analyzed data to generate meaningful insights and conclusions.
- Report the findings in a clear and organized manner, ensuring that the research objectives are addressed and supported by evidence from the interviews.
- Include direct quotes or examples from the interviews to illustrate key points and provide authenticity to the findings.

Throughout the interview process, it is essential to ensure the ethical treatment of participants, maintain their confidentiality, and obtain their voluntary informed consent. Researchers should also be open to unexpected insights or emerging themes that may arise during the interviews, allowing for flexibility in the questioning and analysis process.

IV. APPROACHES FOR OVERCOMING ISSUE:

A. Continuous curriculum updates:

Continuous curriculum updates are essential in DevOps education to ensure that the curriculum remains relevant and aligned with the evolving needs of the industry. Here are some considerations and strategies for implementing continuous curriculum updates:

B. Industry Engagement:

Foster strong connections with industry professionals and organizations to stay updated on the latest trends, tools, and practices in DevOps.

Engage industry experts in curriculum development and review processes to incorporate real-world insights and ensure industry relevance.

Regularly seek feedback from employers and graduates to identify emerging skill requirements and areas for improvement.

C. Collaboration with Practitioners:

Collaborate with practitioners who have hands-on experience in DevOps to provide guest lectures, workshops, or practical assignments.

Encourage practitioners to share their experiences, case studies, and best practices to enrich the curriculum with real-world examples.

D. Faculty Professional Development:

Provide faculty members with opportunities for professional development and continuous learning in DevOps through workshops, conferences, and industry certifications.

Encourage faculty members to engage in research and industry projects to stay updated on the latest developments in DevOps.

E. Curriculum Mapping and Evaluation:

Regularly assess the curriculum to identify gaps, redundancies, and areas for improvement.

Align the curriculum with industry-recognized certifications or frameworks, such as the DevOps Institute's DevOps Foundation or the DevOps Agile Skills Association (DASA) Competence Model, to ensure industry relevance

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 6, June 2023

and recognition.

Conduct periodic evaluations, surveys, and feedback sessions with students, faculty, and industry partners to gather insights on the curriculum's effectiveness and areas needing improvement.

Integration of Emerging Topics and Technologies:

Stay informed about emerging topics, tools, and technologies in the DevOps landscape, such as cloud-native architectures, Kubernetes, serverless computing, or observability.

Integrate these emerging topics into the curriculum, either as standalone modules or as part of existing courses, to expose students to cutting-edge practices and prepare them for industry demands.

A. Agile Curriculum Development:

Adopt an agile approach to curriculum development, allowing for flexibility, rapid updates, and iterative improvements based on feedback and industry changes.

Embrace continuous feedback loops with students, industry partners, and faculty members to address emerging needs and ensure ongoing curriculum relevance.

Continuous curriculum updates enable educational institutions to equip students with the latest knowledge, skills, and practices in DevOps. By actively engaging with industry, practitioners, and students, educational institutions can create a dynamic curriculum that meets the evolving demands of the DevOps field and prepares students for successful careers in the industry.

B. Cross-functional collaboration

Cross-functional collaboration is a critical aspect of DevOps education as it mirrors the collaborative nature of DevOps practices in real-world scenarios. Here are some key points on the importance of cross-functional collaboration in DevOps education:

C. Interdisciplinary Learning:

1. DevOps involves the integration of various disciplines, including software development, operations, quality assurance, security, and more.
2. Encouraging cross-functional collaboration in DevOps education helps students develop a holistic understanding of the different roles and responsibilities involved in the DevOps lifecycle.
3. Students gain exposure to diverse perspectives and learn to work effectively with individuals from different backgrounds, fostering a collaborative mindset.

D. Real-World Simulation:

1. Cross-functional collaboration provides students with an opportunity to simulate real-world scenarios they are likely to encounter in DevOps teams.
2. Group projects and assignments that involve students from different disciplines encourage collaboration, communication, and coordination across functions.
3. Students learn to appreciate the value of each role and understand how collaboration and effective communication contribute to successful DevOps practices.

E. Shared Accountability:

1. DevOps emphasizes shared ownership and accountability across teams rather than siloed responsibilities.
2. In a cross-functional collaborative environment, students learn to take collective responsibility for the end-to-end delivery process, from development to deployment and maintenance.
3. Collaboration encourages individuals to work together to identify and address bottlenecks, resolve conflicts, and optimize processes.

F. Practical Application of Skills:

1. Cross-functional collaboration provides students with opportunities to apply their technical skills in a collaborative setting.
2. Students from different disciplines can learn from each other, share their expertise, and collectively solve problems.
3. By working together on projects, students develop skills in effective communication, teamwork, negotiation, and conflict resolution, which are crucial in DevOps environments.

G. Experiential Learning:

1. Cross-functional collaboration fosters experiential learning by exposing students to a variety of perspectives, challenges, and approaches.
2. Through collaboration, students gain insights into the complexities and nuances of cross-functional teamwork, enabling them to develop adaptability, empathy, and problem-solving skills.

To promote cross-functional collaboration in DevOps education, educators can design collaborative learning activities, group projects, and workshops that encourage students from different disciplines to work together. Creating a supportive and inclusive learning environment, facilitating effective communication, and providing opportunities for reflection and feedback are essential for maximizing the benefits of cross-functional collaboration in DevOps education.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 6, June 2023

V. DISCUSSION

Table1. Participants' Profile (N=14).

Id	Experience teaching (years)	Experience teaching DevOps (years)	Experience in industry (years)	Source
P1	12-17	4-5	7-11	Kaggle
P2	3-7	4-5	—	Kaggle
P3	12-17	3-4	7-11	Personal contact
P4	7-11	4-5	12-17	Kaggle
P5	3-6	3-4	3-7	GitHub
P6	7-11	3-4	---	Alura
P7	17-21	3-4	17-22	Google
P8	7-11	4-5	31+	Kaggle
P9	27-32	4-5	27-33	Personal contact
P10	4-7	4-5	20-24	GitHub
P11	22-28	3-4	13-18	Kaggle
P12	12-17	3-4	18-22	Google
P13	8-12	2-3	5-9	Kaggle
P14	7-12	2-3	—	GitHub

Table 2: Courses' information (N=13).

Id	Interviewee	Course Level	Country	Years of editions of the course
C1	P1	A+	Bahrain	2017–2021
C2	P2	A+	Bahrain	2019–2022
C3	P3	A+	Bahrain	2019–2022
C4	P4	NPTEL	Bahrain	2020–2022
C5	P5	UG	Bahrain	2019, 2020, 2022
C6	P6	NPTEL	Bahrain	2021, 2022
C7	P7	UG	Bahrain	2018–2022
C8	P8	G	USA	2020–2022
C9	P9	G	USA	2018–2022
C10	P10, P14	UG	France	2018–2022
C11	P11	U	Canada	2021, 2022
C12	P12	U	Sweden	2020, 2021,2022
C13	P13	G	Canada	2020–2022

Key: A++ = associate degree; NPTEL = National Programme on Technology Enhanced Learning; UG = undergraduate degree; G = graduate degree.

VI. RESULT

The interviews lasted an average of 32.9 minutes (standard deviation=14.6). We give a collection of artefacts [8] that contain all of the data collected and analysed in this paper. Throughout the interview process, the saturation level of issues and methods. The percentage of difficulties and recommendations identified in earlier interviews is referred to as interview saturation. We called it quits on the interview phase when new interviews only resulted in small adjustments to the topics. This corresponded to a saturation level of around 80%.

VII. CONCLUSION

This research paper will present the findings from the interview study, providing valuable insights into the issue faced in DevOps education and approaches for addressing these issues. By addressing these challenges, educational institutions and training providers can better equip individuals with the skills needed to thrive in the DevOps landscape and contribute to successful organizational transformations.

REFERENCES

[1] Sjøberg, D. I. K., Dybå, T., & Jørgensen, M. (2016). The future of software engineering education: A roadmap. ACM Transactions on Computing Education (TOCE), 16(4), 19.

[2] Lenarduzzi, V., Taibi, D., Pahl, C., & Janes, A. (2018). Challenges in teaching DevOps: A multi-vocal literature review. In Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training Track (pp. 176-185). ACM.

[3] Chatzoglou, P. D., Efraimidis, P. S., & Vrana, V. (2019). Teaching DevOps in academia: A systematic literature review. Journal of Systems and Software, 149, 12-35.

[4] Gravina, R., Nuseibeh, B., & Perini, A. (2020). DevOps education in software engineering curricula: A systematic mapping study. Journal of Systems and Software, 162, 110513.

[5] Cheung, S. C., Chan, T. W., & Ng, C. F. (2021). Challenges in teaching DevOps: An empirical study of instructors. International Journal of Information and Education Technology, 11(3), 135-140.

[6] Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press.

[7] Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Addison-Wesley.

[8] Sjøberg, D. I. K., Hannay, J. E., Hansen, O., Karahasanovic, A., & Liborg, O. B. (2016). Building Theories in Software Engineering - An Interview Study. Information and Software Technology, 70, 77-93.

[9] Gousios, G., Vasilescu, B., Serebrenik, A., & Zaidman, A. (2015). Lean GHTorrent: GitHub data on demand. Empirical Software Engineering, 20(4), 845-881.

[10] Dawei Yang, Daojiang Wang, Dongming Yang, Qiwen Dong, Ye Wang, Huan Zhou, and Daocheng Hong. 2020. DevOps in Practice for Education Management Information System at ECNU. Procedia Computer Science 176 (2020), 1382–1391.