International Journal of Business Quantitative Economics and Applied Management Research
Volume-7, Issue-3, 2022 ISSN No: 2349-5677

THE ROLE OF DEVOPS IN PROGRAM MANAGEMENT BRIDGING DEVELOPMENT AND OPERATIONS EFFECTIVELY

Sandeep K Guduru
USA
gudurusandeep@gmail.com

Abstract

DevOps, which integrates the drive of software development together with operational support, has remained one of the biggest hallmarks in the management of software programs as far as efficiency, agility and continuous development are concerned. This type of review document examines how DevOps has reduced the divide between the elaborate process of SDLC design and operation of IT and the implications of this for program management practices. The implementation of DevOps strategies like CI/CD, IaC and automated testing increases convergence, trims the cycle time to market and enables high level of consistent product delivery. The paper goes further to discuss DevOps solutions to the most common challenges in program management: siloed communication, wasted project management time, and tightening quality controls. Case studies from the industry are analyzed and discussed to explain the benefits of such scenarios as the use of AI in the automation of DevOps processes and the employment of multi-cloud strategies in them. Despite the benefits, the article acknowledges the difficulties of adoption, including talent gaps, organizational opposition, and the technique's complexity. The paper positions DevOps as a new wave in program management and provides practical recommendations; this paper discusses insights into possible future history and explains how to foster innovation and operational effectiveness in the constantly changing digital world.

Keywords: DevOps, software development life cycle, program management, stakeholder engagement, program manager.

I. INTRODUCTION

Program management is a strategic discipline that principally comprises the management of a number of interrelated projects for the purpose of attaining specific organizational goals and objectives. As opposed to project management, which is all about managing the process of delivering individual projects, program management looks at a number of projects as a bigger system and then integrates all of them to offer more collective gains than individual projects on their own [1]. Fundamentally, program management connects to governance, control of resources, risk control, and performance control. It does this by setting out clear objectives, outcomes, and priorities means it helps organizations negotiate through diverse projects, products and systems successfully.

The term "DevOps" refers to a collection of practices that helps operations and developers work together to improve the speed, reliability, and quality of software and service delivery. The term "DevOps" is a combination of the words "Dev" (developers) and "Ops" (operations). The term "DevOps" refers to an approach to software development, deployment, and support that emphasizes collaborative problem-solving within an empowered team [2].

Program managers have the central coordination role in a program since it relates to both the organizational strategy and stakeholders, as well as ensuring that delivered projects meet the stakeholders' expectations. Currently, business environments are highly volatile, and being able to respond to new technologies quickly and effectively is crucial for any organization, and that is where program management comes into play. It offers the framework and strategies through which interdependency can be addressed, risks avoided or mitigated, and sustainability achieved [3].

Fig. 1. Overview of program management

Integrating DevOps into program management has also benefit of increasing the speed of delivering a project, in addition to increasing transparency, risk management, and flexibility in managing programs. It connects the technology staff and the business goal while allowing for the delivery of applications fast and accurately and taking into account altering needs and conditions. Through the creation of the DevOps culture, program managers are able to introduce collaboration and feedback as well as receive timely feedback regarding issues that are slowing down a process or tying up essential resources. DevOps automation in the process eliminates issues related to manual handling of tasks like deployment, testing and monitoring in specific cyclic activities [2].

A. Structure of the paper

The following paper is organized as follows: Section II provides the overview of DevOps in program management, Section III provides the leveraging DevOps to revolutionize program management practices, Section IV defines the challenges in bridging development and operations, and Sections V and VI provide the Literature Review on this topic and Conclusion with future work.

International Journal of Business Quantitative Economics and Applied Management Research
Volume-7, Issue-3, 2022 ISSN No: 2349-5677

II. OVERVIEW OF DEVOPS IN PROGRAM MANAGEMENT

DevOps represents the convergence of "development" and "operations," fostering a collaborative environment between traditionally siloed teams. At its core, DevOps is grounded in principles such as continuous integration and delivery (CI/CD), infrastructure as code (IaC), and robust automation and monitoring. These practices focus on improving software development and operational processes as a way of helping businesses get better development cycles, secure and reliable releases and better outcomes. Not only does it involve the use of tools and integrating methodologies, it means that it eliminates most of the human factors in it, and therefore, it is highly efficient [4].

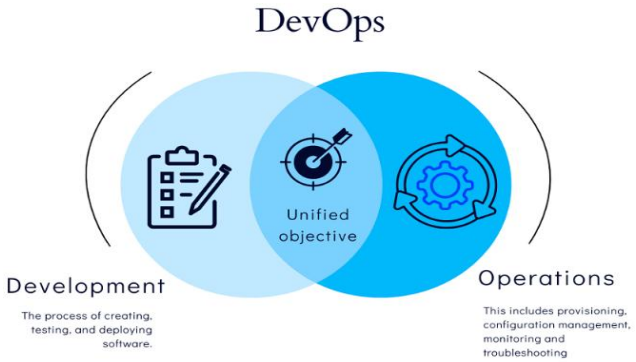

Fig. 2. DevOps in program management

A. Key Aspects of Program Management

This is the role of program management, which is a strategic mechanism linking numerous integrated projects to the overall organizational objectives. Program managers are responsible for areas such as planning, governance, resources and risks [5]. Thus, program management helps to manage resources, as well as time, cost, and quality so that outcomes achieve planned goals and objectives of an organization.

- **Strategic Alignment:** Responsible for ensuring that the program complements the organization's strategic direction and direction. Concerned with the provision of results with wider utility as opposed to the tangible end-products of projects. Helps in the identification of certain projects and resources that need urgent attention on the basis of their relevance to business strategy.
- **Governance and Oversight**: Promotes a systematically organized approach to decision making process throughout the company. Deals with coordination of task, duties and accountability of the parties involved. Subheadings herein are establishment of steering committees or boards to provide directions and deals with any escalated issues.
- **Interdependency Management:** Coordinates and oversees the relationships between projects within the program. It makes sure that working schedule of resources, timelines and deliverables do not interfere with each other. Emphasizes hunting for balance in project work to optimize program benefits.
- **Risk and Issue Management:** Regularly assesses risks and has strategies in place to manage them to reduce their impact on a business. Generates backup strategies in case of disruptions

or the failure of a given project. Manages it in real time so that problems are solved quickly and program stays afloat.

- **Resource Optimization:** Balances the allocation of human, financial, and technical resources across projects. Promotes orderly use of resources so that maximum results are achieved with minimum resources. Mitigates capacity issues and resource utilization issues in one interrelated plan.
- **Stakeholder Engagement:** Gains support from other stakeholders by setting up understanding and ensuring that all possess similar expectations. Serves as a tool for reporting progress made in the program to the intended stakeholders. Co-ordinates stakeholders' opinion and feedback, to maintain program objectives are still realistic and feasible.

B. Synergies Between DevOps and Program Management

DevOps and program management both have the primary goal of successfully delivering value and quality at a low cost. Program management provides the program level management to properly prioritize DevOps initiatives, while DevOps provides the tactical application and execution of DevOps initiatives. Cumulatively, it promote openness, responsibility, and commitment to development, guaranteeing interconnection and cooperation of all teams accomplishing the company goals [6].

- **Infrastructure as Code (IaC):** IaC helps provide and configure infrastructure by code and makes implementation and configuration consistent and idempotent. IaC gives program managers confidence that can standardize environments across projects to save time on setup, and avoid environmental mismatches that could slow delivery [7].
- **Real-Time Monitoring and Analytics**: Modern monitoring tools such as Prometheus, Grafana, and ELK stack give system performance reports in real-time and also users' activity [8]. The present analysis capabilities enhance program management and enable identification of program issues and solutions in real- time.
- **Continuous Feedback Loops**: Through testing and every software release, including deployment, DevOps provides constant feedback. The above feedback loops make it possible for program managers to track project progress in real time, making it easier to adapt to changes and embrace an adaptive process for the attainment of program objectives [9].
- **Containerization and Microservices**: Docker and Kubernetes are the main tools for the implementation of such containerization, enabling applications to be designed, deployed, and grown within separate surroundings. Microservices implementation also helps to increase the modularity and flexibility of the given application [10]. Program managers also enjoy modularity since specific program segments can be developed and deployed while minimizing the extent of dependency among the segments or successive segments of the program.
- **Collaboration and Version Control Platforms**: Website development tools include Git, Github, Gitlab, and Bitbucket that maintain version control for development and operation crews. It assists the program managers in supervising the technical development, monitoring changes, and keeping records all in view, synchronizing the collaborative endeavors with the program goals and outcomes.

- **Cloud Integration and Scalability:** DevOps integrates with popular cloud platforms such as AWS, Azure, Google Cloud and provide instant resource availability, scalability, and reproducibility. This is because integration to the cloud enables program managers to fulfill the needs of growing and fluctuating projects to enhance scalability and affordability. Cloud-native solutions decrease the time to market and help in utilizing resources in most projects [11].

III. LEVERAGING DEVOPS TO REVOLUTIONIZE PROGRAM MANAGEMENT PRACTICES

Modern program management is characterized by rapid development and is currently exposed to high demand for speed, quality, and flexibility. Legacy program management methodologies are inadequate to address these challenges more so in the technology-intensive fields. DevOps, being an enabler, brings agility, transparency, and operational efficiency to the program management. This section goes further to show how DevOps can help improve program management and how the concept can solve problems of program execution to produce more positive results in the long run.

Fig. 3. Leveraging DevOps to Revolutionize Program Management

A. Facilitating Cross-Functional Collaboration

This means that the DevOps concept is aimed at removing the barriers between the teams to enhance cross-collaboration. DevOps brings together development, operations, as well as other stakeholders into one cohesive process to be on the same page. Scrum, daily meetings, and retrospectives make communication more open and credible and lessen the time to make important decisions [12]. These collaborations are crucial when it comes to sustainable implementation of huge reforms.

B. Enhancing Continuous Delivery and Deployment

CI/CD are two core practices of DevOps that concern continuous integration and continuous delivery of code changes. For program management, CI/CD gives the teams an ability to have set releases per a given period and ensure it respond to shifting market needs promptly [13]. Interactive and continuous integration and development controls risks and increases speed-to-market, which is commensurate with program goals.

C. Improving Transparency and Real-Time Monitoring

Realtime monitoring, logging and analytical tools seamlessly delivered by DevOps allow for visibility of system health and project status. To program managers, these capabilities enhance the identification and management of risks and also enhance the communication channel between the program and its stakeholders [14]. Real-time data enables teams to respond to challenges proactively and thereby avoid compromise on project and organizational objectives.

- **Automation in DevOps:** An Essential in Implementing DevOps Automation is at the center of DevOps as a way of eliminating repetitive tasks from the process. The key automation practices involving R&D teams are test automation and getting automation of deployment around infrastructure. For program management, automation makes work standardized across projects and also reduces the operational overhead and increases scalability. By automating some key tasks, it becomes easier for various teams to work on initiatives that add value [15].

D. Popular DevOps Tools Supporting Program Management

Several DevOps tools exist for program management to make work more efficient and hassle-free by providing automation. CI/CD tools like Jenkins, GitLab, and CircleCI streamline software development and deployment. Application orchestration solutions are also available with Kubernetes and Docker to help oversee the resources at the container level. Terraform and Ansible help to automate the process of creating the infrastructure, while tools such as Prometheus, Grafana, and Splunk are used to monitor the performance of the created system.

E. Role of CI/CD Pipelines in Program Success

Continuous integration and continuous delivery are part of DevOps as that create a reliable path for software developing, testing, and delivery. CI/CD pipelines execute these fatal undertakings autopilot by eliminating human errors, shortening the delivery cycles, and enhancing the software quality. To the program managers, these pipelines provide a way of monitoring developmental processes to avoid or manage risk factors hence allowing for achievement of development objectives [16]. The consistency and reliability of CI/CD pipelines prove them to be significant when it comes to program predictability.

F. Integrating DevOps Practices into Program Management

Frameworks for organizations to fully benefit from DevOps, their practices must have to fit into existing program management frameworks. It means that work processes and tools have to be also aligned with organizational goals and such as methodologies as sprint planning or retro. Company stakeholder reviews, together with continuous improvement activities, check on progress to ensure that DevOps practices serve the purpose of enhancing program effectiveness in the long run.

IV. CHALLENGES IN BRIDGING DEVELOPMENT AND OPERATIONS

Addressing the gap between these two vital areas of practice calls for cultural and philosophical transformations as well as technical and operational refinements. To that end, it is important to understand that there are numerous challenges that organizations face while trying to adopt

DevOps, including change management resistance, lack of skills, and automation challenges. It is thus important to develop awareness of these challenges for purposes of developing ways of addressing them. Understanding these challenges enhances focus on the key areas that should be addressed to create better relationships between development and operations teams.

A. Common Silos in Traditional Approaches:

In other traditionally developed IT structures, the development and operations crews work separately, which results in sub-optimality and undesirable goals. While developers are still working on their skills to set new features as quickly as possible, operations teams are interested in system stability and readiness. This causes a number of narrow points, most especially during the deployment phases and will generally lead to poor overall efficiency.

B. Misalignment of Goals Between Development and Operations:

Another major problem of integration is that the goals of development and operation may be quite different from each other. In development teams, there is much emphasis on the delivery of innovations as quickly as possible, meaning that teams may be overly focused on velocity. This can cause delay, increased cost and low customer satisfaction among organizations and consumers. Solving it means establishing clarity on goals and positive approaches that will sustain innovation and stability at the same time.

C. Resistance to Change:

Organizational and Cultural Barriers DevOps mean a change of organizational culture and the organization's activities. Sometimes, employees are just not willing to change and cannot recognize the opportunities that DevOps offers. Employees may resist implementing changes to tools, processes and ways of working in teams. To overcome this, there must be heightened leadership, training and messages about the need to overcome resistance to willingly acceptance of DevOps and the benefits it has to offer to specific teams and the entire organization.

D. Toolchain Integration:

There are gaps between Dev and Ops in coded, tested, and deployed and monitored systems and tools are different. Cohesive and complementary tools otherwise create incongruity, slow production, and hamper automation. It creates the ability to have coordinated DevOps toolchains or incorporate existing tools with the help of APIs.

E. Skill Gaps and Training:

DevOps involves a highly technical support system that calls for quite technical skills that employees usually do not possess in equal esteem in development, operations, and automation. Lack of skilled personnel hinders the imposition of DevOps approach and results in self-relying on certain team members. By offering courses and seminars on a regular basis, accreditation and practical applications give the teams what they need.

V. LITERATURE REVIEW

In the section provide the previous research on the Role of DevOps in Program Management Bridging Development and Operations Effectively.

In, Senapathi, Buchan and Osman (2018) summarizes studies that looked at the variables that affected the adoption of DevOps. It details the results of a comprehensive exploratory case study that looked at how a product development company in New Zealand used DevOps. The research included in-depth interviews with six seasoned software engineers who were asked to track and reflect on the adoption of DevOps methods and concepts over time. Increased deployment frequency was one of the many notable benefits that this case study saw as a result of implementing DevOps principles. For DevOps to provide its promised benefits, they discovered that certain technology enablers—including an automated pipeline and cross-functional organizational structures—were necessary [17].

In, Mishra and Otaiwi (2020) with the objective of studying how DevOps elements affect software quality. Additionally, DevOps might mean a shift in company culture that prioritizes bridging the gap between development and operations. While there are numerous advantages, such as improved quality, to using DevOps, there are also certain difficulties that businesses face. The study's findings shed light on the impact of DevOps on software quality, which is useful for scholars and professionals in the field. According to the results, the primary areas of interest for DevOps research were automation, culture, continuous delivery, and rapid feedback. Many parts of DevOps require additional exploration [18].

In, Lwakatare et al. (2019) gives in-depth accounts of DevOps's practical implementation. The advantages of DevOps, including as rapid releases and few deployment failures, prompted a multiple-case study in five distinct development environments where the practice has been successfully implemented. Before conducting a cross-case synthesis, data was separately coded for each example according to a set of established themes. In particular, for SMEs, their research adds to what is already known about the DevOps framework, its practices, and the effects that its proponents claim it has. They go over two real-world consequences of the findings [19].

In, Nurullah et al. (2018) contributes to software development culture by presenting design analysis as a cornerstone of software development, in conjunction with DevOps and SOA. One outcome of the strategy design study is the establishment of a culture of software development that can streamline the process by reducing the frequency of miscommunication between developers and system users. On many occasions, developers and consumers in need of an application failed to communicate effectively, leading to software development timelines that did not conform to the original plan[20].

In, Raj and Sinha (2020) shows that the project management techniques are affected by the effects of Agile and DevOps methodologies on scope management, quality management, and estimating. Additionally, the literature study makes clear that DevOps and Agile approaches affect team organization via feedback, automation, and shared accountability. Moreover, this

paper offers suggestions to industries who are looking to improve their project management processes and team structure by using Agile and DevOps approaches. The project's success is dependent on the project management procedures. Agile and DevOps are fundamentally altering the norms for the delivery and execution of software projects [21].

In, Sachin, Vernekar and Kumar Dhar (2018) seeks to comprehend the effects, if any, of DevOps on project management approaches as they pertain to the administration and execution of DevOps Projects. In order to accomplish this, information technology managers and devops engineers were asked to fill out a questionnaire. The article concludes with some recommendations for project managers on how to best oversee DevOps initiatives. Efficient use of time and resources is the primary goal of project management approach. It is a collection of procedures and standards for properly managing a project so that it produces the expected results [22].

In, Perera, Silva and Perera (2017) investigated the effects of DevOps on software quality. To better understand the state of DevOps in the business world, a literature review was conducted. The linear regression analysis is used to construct a linear model. To get the hard numbers, they conducted an online survey; to get the soft numbers, they spoke with DevOps and quality assurance specialists to find out how to make software better through DevOps. Feedback from interviews and testing of hypotheses using regression analysis form the basis of the recommendations. The quantitative analysis found that when developers and operations teams use the CAMS framework to perform DevOps, software quality improves. If you want better software, automation is a must [2].

Table I provides a concise summary of each study, including its focus, methodologies, key findings, challenges, and areas for future research. Let me know if you'd like to refine or expand any specific section.

Table I. Literature Review Summary for Devops in Program Management

Reference	Focus On	Key Findings	Challenges	Limitations and Future Work
[17]	DevOps implementation in organizations	Increased deployment frequency through DevOps practices; role of automation pipelines and cross-functional structures	Gradual adoption requires continuous monitoring and reflection	More studies needed to generalize findings to diverse industries and geographies
[18]	Implications of DevOps features on software quality	Enhanced software quality through continuous delivery and fast feedback	Adoption of DevOps brings organizational and cultural challenges	Further research is needed in underexplored areas of DevOps

[19]	DevOps implementation in small/medium companies	Achieved benefits like quick releases and minimal deployment errors	Limited focus on larger enterprises	Insights mostly applicable to small and medium-sized organizations; need further exploration for larger enterprises
[20]	Collaboration between DevOps and SOA in development	Minimized communication errors between developers and users, leading to accelerated development	Timeline mismatches and miscommunication between stakeholders	Exploration of scalability of DevOps-SOA integration
[21]	Impact of Agile and DevOps on project management practices	Enhanced project management through shared responsibility, automation, and feedback	Integrating methodologies into existing processes	Recommendations need validation through industry case studies
[22]	DevOps impact on project management methodologies	Suggestions for maximizing resource and time efficiency through DevOps practices	Lack of practical guidance for project managers	Broader empirical studies are required for validation; focus on real-world implementation challenges
[2]	DevOps impact on software quality	Improved software quality through CAMS framework; automation identified as the most critical factor	Inconsistent implementation of automation practices	Recommendations based on limited datasets; need for more comprehensive studies on automation's impact on broader quality metrics

VI. CONCLUSION AND FUTURE WORK

DevOps has become another great revolution in program management since it helps to address one of the major challenges in an organization: the division between development and operations teams. The changes that flow from the ethic create a culture of collaboration, demand automation, and work on constant improvement That addresses issues such as poor alignment and disconnection between administrative activities, communication silos, and project time lags. Hypothetical practices such as CI/CD, IaC, and testing help to meet the goals of Clients, thus improving the organization's outcome. Effective incorporation of DevOps in program management enhances organizations' ability to support technical methodologies that provide stakeholders with value while being responsive to change in today's business environment. As long as those barriers, including opposition to change, lack of skills, and technical difficulties, persist, the advantages of DevOps, which comprises lower time to market, integrated teamwork, and effective risk handling, prove the importance of DevOps in modern program management.

In future works, DevOps will be driven by emerging technologies. Both AI and machine learning can apply on decision makings of large organizations and DevOps in multi-cloud environment will solve scalability and flexibility. In addition, there is scope for future research in the development of solutions to cultural and organizational barriers as it applies to DevOps. These areas will have to be further investigated in the course of the proposed work with a view to improving the efficiency of the DevOps initiative, and its ability to address emerging program management needs in a globalizing and rapidly digitalizing world.

REFERENCES

1. M. Miterev, M. Engwall, and A. Jerbrant, "Exploring program management competences for various program types," Int. J. Proj. Manag., 2016, doi: 10.1016/j.ijproman.2015.07.006.
2. P. Perera, R. Silva, and I. Perera, "Improve software quality through practicing DevOps," in 17th International Conference on Advances in ICT for Emerging Regions, ICTer 2017 - Proceedings, 2017. doi: 10.1109/ICTER.2017.8257807.
3. C. A. Cois, J. Yankel, and A. Connell, "Modern DevOps: Optimizing software development through effective system interactions," in IEEE International Professional Communication Conference, 2015. doi: 10.1109/IPCC.2014.7020388.
4. N. Forsgren, D. Smith, J. Humble, and F. Jessie, "Accelerate State of DevOps Report 2019," 2019.
5. D. M. Malikova, "Key aspects activation of program-project production management at enterprises of the Russian Federation's defense-industry complex," Econ. Ind., 2018, doi: 10.17073/2072-1633-2018-1-20-28.
6. H. L. Truong and P. Klein, "DevOps Contract for Assuring Execution of IoT Microservices in the Edge," Internet of Things (Netherlands), 2020, doi: 10.1016/j.iot.2019.100150.
7. A. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "A systematic mapping study of infrastructure as code research," Inf. Softw. Technol., 2019, doi: 10.1016/j.infsof.2018.12.004.
8. R. Gunasekaran, S. Oral, D. Dillow, B. Park, and G. Shipman, "Real-Time System Log Monitoring/Analytics Framework," Cug.Org, 2015.
9. A. Mousavi, C. Mares, and T. J. Stonham, "Continuous feedback loop for adaptive teaching and learning process using student surveys," Int. J. Mech. Eng. Educ., 2015, doi: 10.1177/0306419015606618.
10. N. D. Keni and A. Kak, "Adaptive Containerization for Microservices in Distributed Cloud Systems," in 2020 IEEE 17th Annual Consumer Communications and Networking Conference, CCNC 2020, 2020. doi: 10.1109/CCNC46108.2020.9045634.
11. Z. Hossain et al., "Design Guidelines and Empirical Case Study for Scaling Authentic Inquiry-based Science Learning via Open Online Courses and Interactive Biology Cloud Labs," Int. J. Artif. Intell. Educ., 2018, doi: 10.1007/s40593-017-0150-3.
12. K. H. Tsai and T. T. Hsu, "Cross-Functional collaboration, competitive intensity, knowledge integration mechanisms, and new product performance: A mediated moderation model," Ind. Mark. Manag., 2014, doi: 10.1016/j.indmarman.2013.08.012.
13. J. Wettinger, U. Breitenbücher, and F. Leymann, "ENHANCING CLOUD APPLICATION DEVOPS USING DYNAMICALLY TAILORED DEPLOYMENT ENGINES," Serv. Trans.

Cloud Comput., 2016, doi: 10.29268/stcc.2016.0002.
14. I. Karamitsos, S. Albarhami, and C. Apostolopoulos, "Applying devops practices of continuous automation for machine learning," Inf., 2020, doi: 10.3390/info11070363.
15. H. Huang, H. Zhang, and D. Shao, "Practical Impacts of Automation Tools in Support of DevOps in China," Ruan Jian Xue Bao/Journal of Software. 2019. doi: 10.13328/j.cnki.jos.005788.
16. A. F. Nogueira, J. C. B. Ribeiro, M. Zenha-Rela, and A. Craske, "Improving la redoute's CI/CD pipeline and devops processes by applying machine learning techniques," in Proceedings - 2018 International Conference on the Quality of Information and Communications Technology, QUATIC 2018, 2018. doi: 10.1109/QUATIC.2018.00050.
17. M. Senapathi, J. Buchan, and H. Osman, "DevOps capabilities, practices, and challenges: Insights from a case study," in ACM International Conference Proceeding Series, 2018. doi: 10.1145/3210459.3210465.
18. A. Mishra and Z. Otaiwi, "DevOps and software quality: A systematic mapping," Computer Science Review. 2020. doi: 10.1016/j.cosrev.2020.100308.
19. L. E. Lwakatare et al., "DevOps in practice: A multiple case study of five companies," Inf. Softw. Technol., vol. 114, pp. 217–230, 2019, doi: 10.1016/j.infsof.2019.06.010.
20. F. Nurullah, G. Wang, E. R. Kaburuan, and A. N. Fajar, "The Collaboration of DevOps Automation and SOA to Accelerate Software Development Culture," 1st 2018 Indones. Assoc. Pattern Recognit. Int. Conf. Ina. 2018 - Proc., no. September, pp. 262–266, 2018, doi: 10.1109/INAPR.2018.8627022.
21. P. Raj and P. Sinha, "Project management in era of agile and devops methodlogies," Int. J. Sci. Technol. Res., 2020.
22. P. Sachin, S. Vernekar, and S. Kumar Dhar, "Impact of DevOps Skills on Project Management Outcome an Empirical Study," Eur. J. Mol. Clin. Med., vol. 7, no. 8, pp. 5282–5291, 2018.