DevOps with Agile: Best Practices to Improve Software Quality

Saifullah Adnan, Khadeeja Moin*, Jaweria Imran
Center for Computing Research Department of Computer Science & Software Engineering,
Jinnah University for Women, Karachi, Pakistan
*E-mail: khadeejamoin@gmail.com

ABSTRACT

The emphasis on software quality is increasing as it is becoming more of a success indicator for a software product. Therefore, a lot of strategies are being utilized to optimize the process and achieve the best possible quality. Agile methodology fulfills almost every criteria of a better quality product by improving the overall process of software development. This study revolves around exploring the DevOps model and key practices to improve software quality. The paper helps in distinguishing the Agile methodology from DevOps model and highlights their benefits. The researchers have used the novel method of systematic literature review to carry out this study. The paper investigates the superiority of DevOps as compared to Agile and unveils that the combination of both should be implemented to achieve top-notch results. The paper also outlines some guidelines to improve software quality using DevOps as well as a few challenges of adopting this model. The overall findings state that Agile and DevOps function complementary to each other. Also, DevOps should be adopted as an organizational culture rather than a technique. It helps improve the overall process efficiency and reduces cost by enhancing collaboration and task automation.

Keywords: Agile, DevOps, DevOps challenges, Software quality

INTRODUCTION

In the software development process, it is crucial to follow a standard model/ technique to achieve the desired results. Agile methodology combines the incremental and iterative development approach and follows the following parameters: to plan adaptively, to develop and evolve product in an iterative manner, to respond positively to changing requirements, and to improve communications (Kumar & Bhatia, 2012). The novelty of agile methodology is indicated by its work practices. The major concern in agile is to prioritize the expertise and skillset of software developers against the managerial bodies (Zaitsev et al., 2020). Another prime focus is to receive input and feedback from associated stakeholders, customers, and end-users throughout product development process (Cohn, 2004). According to researchers in (Wang & Liu, 2018), agile methodology emphasizes only the part of software development while leaving a wide gap in deployment and operations phases.

To eliminate this gap, software industry tends to switch from traditional development models towards *"DevOps"* to deliver high-quality products and services as quickly as possible (Battina, 2021). To enhance the productivity of software development process, DevOps provides strategies to improve collaboration between Development and Operations teams (Sugumaran, 2008). DevOps refers to a collaborative strategy between the development and operations teams in order to reduce the time required to reflect changes in production environment while maintaining product quality (Zhu et al., 2016). Adopting DevOps significantly reduces fault occurrence and speeds up the process of releasing upcoming features (Mohammad, 2018). DevOps helps in accomplishing a quicker cycle of developing and deploying small-scale user stories within a matter of days (Dörnenburg, 2018). The study in (Bucena & Kirikova, 2017) suggests that DevOps optimizes the

communication between Development and Operations departments and uses automation to significantly lessen the number of required resources. In the past few years, DevOps has emerged as a driving force for the software industry by delivering reliable and high-quality services at a much faster rate than ever before (Mishra & Otaiwi, 2020). The above aim can be achieved by implementing a continuous delivery and development lifecycle by adopting agile methodology (Senapathi et al., 2018). Using Agile and DevOps in proper alignment provides better outcomes. When integrated together, both technologies tend to ensure fast product delivery as well as a customized communication environment (Almeida et al., 2022). This paper unveils the Agile and DevOps technologies and explores the effect on outcomes when applied together.

Related Work

Authors in (Almeida et al., 2022) conducted a study to investigate the merits of implementing Agile and DevOps in parallel. The study implies the notion of Agile methodology as a solution to all the limitations of using traditional development models. The key characteristics highlighted in the study include quicker delivery of software artifacts to promote continuous delivery cycle, flexibility for projects with changing and complex requirements, and constant feedback. The emergence of DevOps technology aims to elevate the Agile approach to a greater extent. DevOps revolutionizes the traditional practices of product development by introducing an atmosphere of collaboration by combining the Development and Operations teams. The combination of both teams ensures the factors of stability and constancy in the production phase along with automating the multiple activities involved in transferring the developed code over to the operations team. The authors have also identified a few problems with Agile methodology and also presents relevant solutions offered by DevOps. The researchers incorporated qualitative data analysis techniques including thematic analysis and investigated twelve scenarios where Agile and DevOps had been utilized in combination with each other. The findings reveal several advantages of incorporating both techniques together. Some of these may vary based on the context of case study. However, improved communication, reduced product delivery time, and automation of activities are declared as the fundamental benefits of integrating Agile methodology and DevOps.

The study conducted in (Battina, 2021) discloses the obstacles faced while implementing DevOps strategy in software development. Furthermore, it highlights necessary prevention mechanisms to enjoy maximum benefits. To indicate a clear distinction between Agile and DevOps, authors referred to Agile as a methodology and DevOps as a framework. The major notion of Agile to ensure quick delivery often poses significant challenges to security team. The first challenge in the DevOps culture is that developers are dedicated to producing code in rapid cycles whereas security team is concerned more about system safety which ordinarily results in delayed delivery cycle. The security team requires to carry out an extensive inspection to eliminate any possible vulnerability which can affect a system's operation. However, to keep pace with the development team, the security checks are often compromised. This arise as another challenge of adopting DevOps. The respective mitigation strategies suggest that a set of security criterion should be established and crucial security practices such as implementing firewalls, restricted access control, and code review should be incorporated throughout the process by DevOps teams. Furthermore, the study emphasizes that all teams should work collectively for developing a secure product rather than enforcing entire responsibility on security team. This idea is termed to as *"DevSecOps"* which proposes to integrate security practices in an organization's culture where every team agrees on adhering to security procedures and holds equal responsibility for product security.

Since teams generally work on each component in separate cycles rather than as a single unified group, incremental delivery sometimes results in severely scattered output rather than helping products reach the market more quickly. With knowledge of this drawback, DevOps, a better methodology for producing better goods quickly, is put into use. DevOps is a never-ending cycle of planning, coding, building, releasing, deploying, operating, monitoring, and returning to planning. The authors in (Leite et al., 2019), explore and

talk about DevOps difficulties from the perspectives of engineers, managers, and researchers. They also recommend adopting a microservice architecture for continuous delivery and a cloud services architectural pattern to help with application deployment and operation for engineers at the end of the study. The implementation of lean principles, developer training, and preserving a trustworthy relationship with clients are some implications outlined for managerial levels. And for the academic community, some potential study topics were identified, including software design in the context of DevOps, how well DevOps functions in an IoT, and examining deployment strategies that can assist engineers in selecting the optimal toolkit.

OBJECTIVE

The objective of this study is to identify the best practices to adopt DevOps and Agile together for the improvement of software quality and investigate the major issues while adopting DevOps in organizations environment.

RQ 1: How is DevOps superior to Agile?

Agile methodology is introduced to resolve the issues associated with waterfall model. It promoted incremental development to provide support for new features in every release cycle. However, this methodology suffers from lack of communication and inability to meet deadlines. As the product is continuously developed therefore; it becomes difficult to measure performance and reach a finite end-point. Due to frequent changes and customization, budget parameters can often be disturbed. Moreover, the product can be unstable because features developed in different cycles may not support each other.

DevOps culture overcomes the drawbacks of Agile methodology and works for process optimization. Rather than insisting on quick feature development as in Agile, DevOps prioritizes the automation of processes and functional efficiency. DevOps introduces an organizational culture of collaboration, balance, and provides infrastructure while ensuring cost reduction. Since the teams collaborate and establish unified goals to accomplish, the divided development of product is still compatible. A variety of tools is put into use to automate the process and test the stability and quality of every feature prior to release. All of these practices contribute in meeting time and cost constraints along with enhancing the transparency of development process.

RQ 2: What are the ideal DevOps techniques for enhancing software quality?

The DevOps culture promotes the environment of collaboration between teams where each team is identically responsible for quality and stability of end-product. To enhance the overall product quality, a strengthened bond of continuous feedback between development and operations plays a major role. Authors in (Gottesheim, 2015) suggest that a mutual understanding of quality criteria and performance measurement assists in getting desired outcomes. The study also indicates that automated tools for performance monitoring should be utilized to convey a better understanding of system performance under stressful environments to the developers. This assists developers in interpreting the operation's team perception of quality; thus reduces rework. Furthermore, automating procedures enhances system security by eliminating the chances of human error (Battina, 2017). Increasing the release and deployment frequency is also a key indicator of improving quality through DevOps. Thus adopting major DevOps practices of establishing organizational culture, immediate feedback, collaboration, automation, following a cycle of continuous delivery and deployment, and understanding a common performance criteria can lead to multiple quality attributes including usability, efficiency, maintainability , reliability, and security (Mishra & Otaiwi, 2020).

RQ 3: What are the challenges encountered while utilizing DevOps?

Shorter iterations are supported by DevOps by merging the development (Dev) and operations (Ops) teams, changing organizational structure and culture, automating processes, and introducing new tools. But while

adopting a DevOps, DevOps teams also faced various difficulties that slowed down the process by impeding DevOps enablers or raising the possibility of failing to meet the objectives. Some of the major challenges are described below according to our understanding.

Technical Challenges

Having employees with the requisite technological abilities this problem involves both finding new employees with the necessary technical skills and retraining and keeping on board current employees. Because the key competencies are lacking when needed, a lack of adequately skilled personnel may cause the adoption of DevOps to delay. According to the authors of (Senapathi et al., 2018), finding qualified candidates is the biggest difficulty because the required skill set is scarce. Effective DevOps programs are difficult to implement, however this may be due to the underutilized operational skills and courses in software engineering professional education. In addition, there are a profusion of DevOps technologies accessible, and they are constantly improving. It can be challenging to decide which tools to thoroughly explore, which tools to become familiar with, which tools to try, and which tools to avoid due to the lack of professionals in this field. Such decisions must be supported by a conceptual examination of the needs and insights of the individual and the organization.

Deployment Challenges

DevOps deployment in business is still difficult in addition to other difficulties. Developers and operators work together to drive DevOps. In this method, developers' and operators' duties are intertwined. When new roles within the business are not clearly defined, there is a significant risk of employing this technique, which could result in conflict between developers and operators. Product teams are also known as cross-functional teams. Professionals who are "T-shaped," or have specialization in a small number of domains and foundational skills in several related ones, are drawn to this term “Cross Functional” (Senapathi et al., 2018). As per this paradigm, at least one team member must be an expert in operations. According to this viewpoint, a "DevOps engineer," sometimes known as a "full stack engineer," is a developer with operations expertise. Moving operators to product teams can be all it takes to adopt cross-functional teams. However, pushing engineers to understand operations may be unproductive because they are already overburdened with other knowledge gaps. Making sure that experts communicate with their peer groups to share advancements in the field presents an additional difficulty for cross-functional teams.

Culture Challenges

It is well known that changing a company's culture is not something you can do at will. You can change the culture and affect it, and while it will eventually change, it is very difficult to simply tell everyone to believe and act differently about a certain situation. One of the most important issues that DevOps aims to solve is organizational and cultural constraints (Kotagi & Pareek, 2016). It is extremely difficult to accomplish the shared objective of swiftly deploying value-added software when development and operations are segmented in their own teams or "some loosely connected working groups." Dev and Ops could not speak the same language when expressing their thoughts and problems because they are on different teams. On both sides and from a business perspective, this causes miscommunication, confusion, and additional work. The last challenge is fear: negative attitudes toward sharing knowledge and achieving a common objective coincide with ingrained behaviors and the worry of losing control, name, or authority.

METHODOLOGY

We opted to use the systematic literature review (SLR) method for this proposed study. The systematic literature review (SLR) is a sort of literature review that gathers and critically evaluates a number of research works or papers. A SLR's goal is to offer a thorough review of all the literature that is available and related to a specific research question. To find the most pertinent studies on *DevOps with Agile: Best Practices to Improve Software Quality* in the last ten years, we first looked through a variety of research studies that have been conducted. After identifying the most prominent questions in the context of DevOps and Agile, we then searched various digital databases, including IEEE Xplore and Google Scholar. Upon locating various studies, we choose the one that best answers our study questions, which are: How is DevOps superior to Agile? What are the ideal DevOps techniques for enhancing software quality? and What are the challenges encountered while utilizing DevOps?

CONCLUSION

The prime intention behind the study is to investigate the Agile and DevOps methodology, best practices, and impact on software quality. A systematic literature review of papers from various databases is conducted and analyzed to achieve the results. The findings state that even though Agile is one of the great techniques for improving software quality but still there is room for further improvement. To fill in this gap and to bring enhancement in the overall business process, DevOps model is acquired. DevOps brings in the culture of collaboration and automation within the organization's environment and increases the deployment frequency. It is also observed that the loop of continuous feedback and mutual performance measurement metrics leads towards better quality products and cost reduction. Moreover, a combination of Agile and DevOps can play a vital role in the development of a product according to ideal standards of process development and product quality. The implementation of DevOps is positively related to the software quality.

REFERENCES

1. Almeida, F., Simões, J., & Lopes, S. (2022). Exploring the Benefits of Combining DevOps and Agile. Future Internet, 14(2), 63.
2. Battina, D. S. (2017). BEST PRACTICES FOR ENSURING SECURITY IN DEVOPS: A CASE STUDY APPROACH. International Journal of Innovations in Engineering Research and Technology, 4(11), 38-45.
3. Battina, D. S. (2021). The Challenges and Mitigation Strategies of Using DevOps during Software Development. International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.
4. Bucena, I., & Kirikova, M. (2017). Simplifying the DevOps Adoption Process. BIR Workshops,
5. Cohn, M. (2004). User stories applied: For agile software development. Addison-Wesley Professional.
6. Dörnenburg, E. (2018). The path to devops. IEEE Software, 35(5), 71-75.
7. Gottesheim, W. (2015). Challenges, benefits and best practices of performance focused DevOps. Proceedings of the 4th International Workshop on Large-Scale Testing,
8. Kotagi, M., & Pareek, P. K. (2016). Survey on Challenges in DevOps. International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN, 2347-5552.
9. Kumar, G., & Bhatia, P. K. (2012). Impact of agile methodology on software development process. International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(4), 46-50.
10. Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. ACM Computing Surveys (CSUR), 52(6), 1-35.
11. Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. Computer Science Review, 38, 100308.
12. Mohammad, S. M. (2018). Improve software quality through practicing devops automation. Sikender Mohsienuddin Mohammad," IMPROVE SOFTWARE QUALITY THROUGH PRACTICING DEVOPS AUTOMATION", International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.
13. Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps capabilities, practices, and challenges: Insights from a case study. Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018,
14. Sugumaran, V. (2008). Distributed artificial intelligence, agent technology, and collaborative applications. IGI Global.

15. Wang, C., & Liu, C. (2018). Adopting DevOps in Agile: Challenges and Solutions. In.
16. Zaitsev, A., Gal, U., & Tan, B. (2020). Coordination artifacts in agile software development. Information and Organization, 30(2), 100288.
17. Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and its practices. IEEE Software, 33(3), 32-34.