

DevOps: A Historical Review and Future Works

Mayank Gokarna

IBM India Pvt Ltd, Manyata Tech Park,
Bangalore, India

Raju Singh

IBM India Pvt Ltd, Subramanya Arcade,
Bangalore, India

Abstract- DevOps is an emerging practice to be followed in the Software Development life cycle. The name DevOps indicates that it's an integration of the Development and Operations team. It is followed to integrate the various stages of the development lifecycle. DevOps is an extended version of the existing Agile method. DevOps aims at Continuous Integration, Continuous Delivery, Continuous Improvement, faster Feedback and Security. This paper reviews the building blocks of DevOps, challenges in adopting DevOps, Models to improve DevOps practices and Future works on DevOps.

Keywords- Agile software development; DevOps; Cloud Computing; Automation; Continuous Integration; Continuous Delivery

1. Introduction:

The practices followed during the software development lifecycle play an important role. In the conventional development lifecycle, different teams will play their role at their level. Separate teams make the product life cycle lengthier and also the communication between the teams poor [1]. This kind of development model is called the waterfall model. To break the walls between the teams and to enhance the dissemination of the information the new methodology Agile was discovered. Agile means "to move fast and easy". Agile process methodology improved the interaction between individual teams and improved collaboration. Some of the agile principles are Scrum, Extreme Programming, Lean, Kanban and out of these the Scrum was the first developed principle [2]. Even though the agile process reduced the time of the development life cycle, there were some gaps that needed to be bridged. This is why DevOps evolved and it is the extension of the agile process. DevOps unifies the Development and Operations team. Automation also plays an important role in DevOps. The processes like maintenance and testing were already automated in Agile [3]. DevOps cannot be implemented at one stretch, it needs to be implemented step by step or iteratively. DevOps was divided into 4 areas [4] and it is shown in figure [5]. This paper mainly deals with DevOps and its building blocks, how it's getting improved, adoption methodology, other parameters that help to improve and challenges in implementing the DevOps.

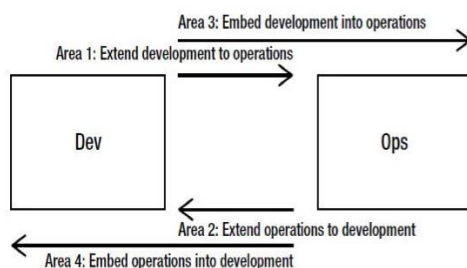


Fig. 1. Areas in DevOps

2. Agile Methodology:

Due to the shortcomings of the waterfall method, the Agile method was introduced. The agile methodology was introduced in the year 2001 [6]. Agile is the evolutionary development model [7] and it aims at continuous improvement of the product features. The agile method integrates the programmers, testers, and QA as the Development team and separate Operations team. Conflicts arise between the Development and Operations teams while deploying the newly developed features and fixing problems [8]. It was hard to maintain the software and to update them whenever necessary [9]. The agile processes are feature and people-centric approaches and so it was a challenge to change from the traditional process-centric approach. In the agile method, the information passing from the Development team and operations team was not faster and not in a frequent manner. This is a bottleneck in the agile method [10]. This gives rise to the new methodology DevOps, which combines the Development and Operations team to enhance communication and frequency.

3. DevOps:

DevOps is a set of procedures that combines the process of Development and Operations [5]. DevOps needs a set of tools to perform the function of combination and integration. In other words, DevOps is a single team that looks after development, testing, and operations. In DevOps, the total product cycle doesn't break at any point [5]. The DevOps has four Dimensions [1], they are

1. Collaboration,
2. Automation,
3. Measurement,
4. Monitoring

DevOps is the extension of the agile method of software development [11]. DevOps focuses on the continuous delivery of the software along with continuous integration. Automation also plays a vital role in reducing the latency of product releases. DevOps not only improves collaboration and communication but also fast and continuous delivery, regular updates, increases reliability, etc [12].

3.1 Continuous Deployment:

Cloud computing plays an important role in the case of Continuous Delivery [11]. The cloud-based tools help to bridge the gaps between the need and delivery and also provide faster feedback. In case of a rapid deployment process, testing should be automated to reduce the latency [13]. In a study [15] done on 19 companies, it was reported that 11 out of 19 companies use the Continuous Deployment strategy. The traditional methods like incremental/ iterative and ad-hoc approaches failed to satisfy the needs of the software firms [16]. In [17], the author discussed how to prepare a firm for the continuous delivery process. In that, it

was reported that continuous delivery should be adopted as small increments and repetitive tasks should be automated. The benefits of adopting continuous delivery are accelerated time to market, building the right product, improved productivity and efficiency, reliable releases, improved product quality and customer satisfaction [18]. The code infrastructure helps in fast releases. Before the emergence of DevOps, there was no shared infrastructure for software development in a firm and so the employees used to work individually in their environment and so the delivery process gets delayed due to increased time for integration.

3.2 Continuous Integration:

DevOps itself represents the integration between Development and Operations teams. DevOps enables the continuous integration of all processes involved in product development and so all the process is done by a single team throughout the cycle [5]. Inadequate communication is the key problem that triggers the need for DevOps to be implemented. DevOps enables the software firm to provide more features and continuously improve it based on the feedback within a short period [19]. DevOps not only integrates the teams, but also the tools used in various stages of product development [20]. The new framework was proposed in [2], to implement the process of continuous integration and deployment over the existing Scrum product cycle. The versions of the software tools should be maintained to be compatible with other tools. The version control strategy was proposed to control different artifacts like Source code, configuration files, deployment scripts and binary code of the application in [21]. Automation is necessary for continuous integration and certain tools like Git repository are used to keep track of the changes [14]. HARNESS is a multi-partner research project, which aims at integrating multiple non-homogeneous resources by using version control techniques [22]. A new tool was proposed in [23] called Filling Gap (FG) tool which bridges Development and Operations teams gap and provides quick feedback on the performance to the Development team.

3.3 Cloud:

Cloud is a key player in the DevOps methodology [24]. The cloud DevOps methodology, tools and culture are shown in figure 2.

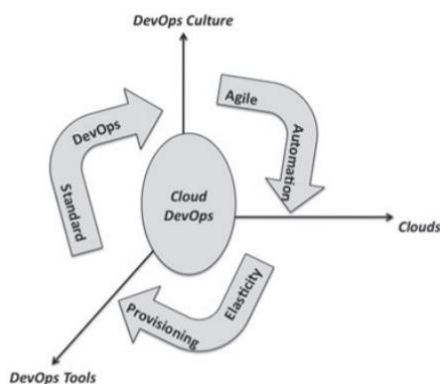


Fig. 2. Cloud DevOps [51]

The cloud ecosystem enables us to have a large number of interconnected components and easy to access and control the components [25] and a simplified cloud ecosystem was proposed.

The serverless model uses Cloud in its architecture [26]. The qualitative data showed that DevOps practices are strongly affected by the new cloud computing model [26]. For cloud applications, lightweight languages were used [27]. Cloud helps in iterative software development, monitoring the running applications on the client-side and to get feedback about the process [20]. The use of a cloud environment also increases the security of the software applications [28]. The DevOps artifacts are classified as Node-centric artifacts and Environmental centric artifacts. The TOSCA was used in cloud applications to make the artifacts interoperable [30].

3.4 Security:

DevSecOps or SecDevOps are the terms associated with Secured DevOps. The continuous security model was proposed in [31], which used open-source software over the cloud to provide security throughout the product development. The security practices used in DevOps [32] are 1) Automation Activities including automated testing, monitoring, code review, software-defined firewall and software licensing 2) Increased collaboration between development and security teams 3) Non-automated security activities are security requirements analysis, performing security configurations, performing security policies, design review, input validation, risk analysis, etc. A Dynamic model was proposed in [33] to rectify the security issues in both the development and operational activities of the multi-cloud application and the proposed model were validated using the real-time application. The framework of the MUSA Dynamic model is shown in figure 3. The cloud security options include serverless computing, Infrastructure as code and security centralization [28].

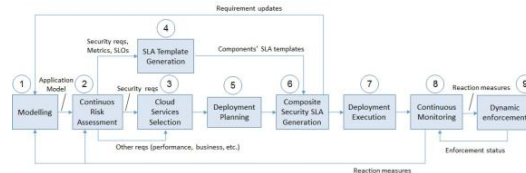


Fig. 3. The framework of MUSA DevOps [33]

It was stated in [28] that cloud makes release faster and more secure. Some of the security activities to be followed in DevOps are Security Requirements Gathering, Threat Modeling, Environment Configuration, Secure Static Analysis, Security-Focused Code Review, Software Penetration Testing, Environment Testing and security Review [29].

3.5 Automation in DevOps:

In case of rapid deployment, manual testing is impossible and so the testing should be automated. The final task in the automation is to check for errors in the system and to rectify them [13]. Apart from testing, automation also helps in scaling of the products [14], the network was formed based on the common services that each system would require and specific tools were used for every task. In the proposed model [34], the workflow is divided into backend and frontend workflow. The proposed model tries to automate the processes that are involved in the development lifecycle. In this model, the cycle of the process starts with gathering data and the gathered data is used to build the application and output of the build are tested and then deployed through the cloud. The automation of the software release causes reduced operational cost, increased productivity, increased accessibility & reliability and optimized performance

[16]. Robotic Process Automation (RPA), is an emerging method to automate repetitive task [16]. Machine learning can be used in automation tasks like testing, quality assurance, fault detection, etc [35]. Automation is the only way to make the release cycle shorter. The architecture was built using reliable existing tools in [24] for automated testing and quality assurance. The building blocks of the proposed architecture are Cloud platform, Virtual management machine and configuration management tool. Not every process can be automated, some processes should be done manually. Automation enables fast feedback from the customer side as well [4].

3.6 Tools used in DevOps:

The tools are required to develop and integrate the processes and it is classified as Build tools and Continuous integration tools [5]. Some of the tools for improving DevOps practices are JIRA, GIT, Jenkins and Docker. The tools should be used based on the hierarchy [36]. The Chef Cookbooks, Puppet Modules, Saltstack modules, Docker images, Juju charms, Bundles and templates are some of the configuration management tools [30]. Earlier mercurial repository was used to store configuration data and to keep track of the changes [14]. The various tools used in various stages of the DevOps cycles are shown in figure 4. Docker is container-based technology and is used to provide isolation between various applications. Using Docker, each application can be configured with a specific version of the Operating system running on the host machine. The end application is delivered as the Docker container [37]. Docker was built on top of the LXC technology, which relies on Linux and hence Docker relies on certain Linux features [38].



Fig. 4. Tools used in DevOps [34]

The project management tool is necessary for an IT firm to manage the data of various projects. One such tool is GZ-Agile Project Management Consolidator, which stores and tracks the activities of DevOps as well as Agile [39]. In a fraud detection system developed by Netfective Technology, SimTool was used to evaluate the metrics [40]. Some of the cloud computing services used in DevOps are Amazon Web Services (AWS), Microsoft Azure, IBM Cloud [34, 27]. Unifying the software development process, the software installation and configuration should not be error-prone. The IT system configuration and management tool are used mostly to enable DevOps, it automates the process of system configuration. The configuration of the system is managed as rules which are organized as modules and classes [41]. The ITIL Framework is also an IT service management tool used in DevOps [42]. The architecture of ITIL was shown in figure 5.



Fig. 5. ITIL architecture [42]

4. Challenges in Adopting DevOps:

The adoption of DevOps in an IT firm is not so easy. In a company it's hard to change the culture of the organization and the practices involved in DevOps will not suit all circumstances [19]. The challenges involved can be categorized as lack of awareness, support, technological feasibility and adapting to the change [43]. Apart from organizational culture challenges, the other challenges include implementing in the existing process, architectural challenges, lack of automation for continuous testing and Legacy systems [18]. The infrastructure also plays an important role in DevOps adoption and it should be compatible and lightweight [44]. The challenges identified using the Fuzzy TOPSIS approach in [45] were Data heterogeneity, data integration, error and inconsistent data, a misspelling in data entry, missing information, traceability of data, Data harmonization, visualization of Data, etc. The conflicts arise between the Development and Operations teams while deploying the newly developed features and fixing the problems. The development team develops a new feature without knowing the problems with the older version. The operations team has to fix the problem with the development team, but the development team is ready to deploy a new One. Some of the obstacles in software development for a highly regulated environment are 1) the operator doesn't know about the code 2) all artifacts are stored in a single isolated container 3) project must be approved before commencement 4) Poor collaboration [52].

5. Proposed Models:

A three-step model for DevOps adoption was proposed in [46]. The model was evaluated by practical implementation. The relationship between the category's agility, automation, collaborative culture, continuous measurement, quality assurance, resilience, sharing and transparency was required for DevOps adoption [46]. The iObserve approach used to tackle the challenges in DevOps adoption was discussed in [47]. The iObserve approach follows MAPE (Monitor, Analyse, Plan, Execute) Control loop. The framework of the iObserve approach is shown in figure 6.

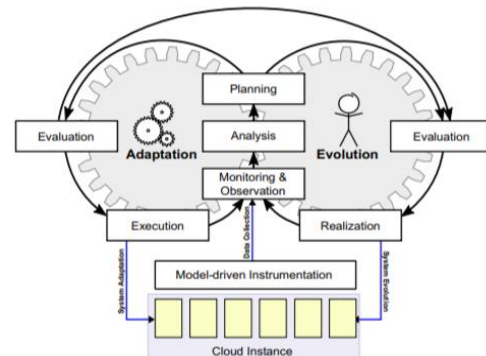


Fig. 6. iObserve model [47]

A technology transfer model shown in figure 7 was proposed in [48] and it was proposed for a scalable multi-party organization, academy and industry, multi-party team. It was reported in [49] that DevOps performance increases with an increase in integrity between DevOps and Knowledge management. Volunteer knowledge sharing should be encouraged in the companies.

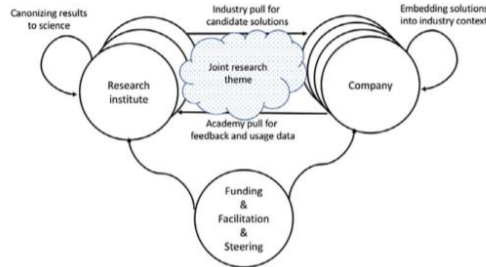


Fig. 7. Technology Transfer Model [48]

A DevOps Maturity model was proposed in [12], with 5 levels of maturity and 4 dimensions of assessment. The proposed model is shown in figure 8 based on the CMMI model.

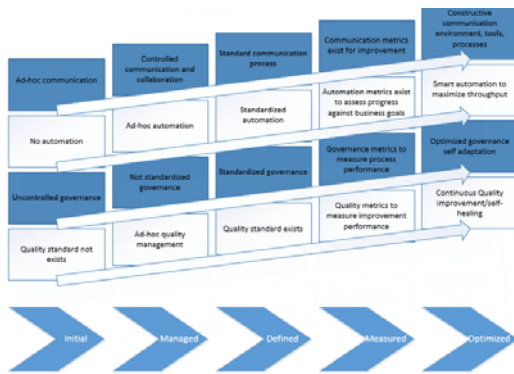


Fig. 8. DevOps Maturity Model [12]

The concept of Composable DevOps architecture was proposed in [50]. The composable nature is to break up things into smaller fragments and then to integrate the entire development pipeline. The composable DevOps is an iterative way to make collaboration between development and Operations.

6. Quality:

In software product development using new techniques, the quality of the developed product should also be maintained properly. The quality assurance of the product is improved by the use of an automated DevOps pipeline. The releases are made more often to improve the features of the product [19]. The set of metrics used to assess and evaluate the DevOps practices were discussed in. The proposed organizational metrics to track the continuous improvement are Truck-Factor, Socio-Technical Congruence, Core-Periphery Ratio, Community Member Turnover and Smelly-Quitters. The Technical metrics are Lines of Code, Coupling Between Object Classes, – Code Change Process, developer-related Factors, Runtime Maintainability Measures and Operations Factors. In a fraud detection system designed in [40], the Simtool was used to evaluate the metrics.

The Fuzzy TOPSIS approach was used for the quality assessment in DevOps [45].

7. Future works on DevOps:

By changing the configuration of the existing model, the upcoming model can be developed as multi-dimensional [34]. The DevOps team will possess desirable qualities or skills, full-stack development, analysis, functional, decision-making, social, testing, and advisory skills [53]. In the future, most IT firms will adopt a continuous deployment strategy [15]. The new metrics will be found to evaluate the quality of the DevOps development cycle [19]. Tools like agile consolidator aimed at improving the quality of the process should be built [39]. The auto-scaling policies in the combination of both container and node level will be developed [35]. Some future works that can be done on working with Docker containers are Networking, Eclipse Plugin, Port management and Clustering capabilities [38]. External clients are eager to collaborate with focused groups inside the solution providing enterprises to develop mixed solution assets in an agile fashion [50].

8. Conclusion:

In this paper, we have discussed the building blocks of DevOps, challenges in adopting DevOps and future works. DevOps is an emerging optimal practice that should be followed in a software development life cycle to increase releases, reliability, faster updates, effective use of customer feedback, increased quality assurance and security. It was clear that DevOps cannot be implemented in an existing pipeline on a single step. It should be implemented in small increments. Automation should be done in the repetitive tasks and at which humans tend to make errors. The cloud is the key player in DevOps, which takes part in integration, continuous delivery, security and in collecting feedback. Many frameworks of the architecture has been discussed and all these models rectified some challenges or improved the existing benefits of DevOps. Apart from all these factors, the team should be skilled and flexible enough to adapt the cultural changes.

References:

- [1] Lwakatare, Lucy Ellen, Pasi Kuvaja, and Markku Oivo. "Dimensions of devops." In International conference on agile software development, pp. 212-217. Springer, Cham, 2015.
- [2] Samarawickrama, Saliya Sajith, and Indika Perera. "Continuous scrum: A framework to enhance scrum with DevOps." In 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 1-7. IEEE, 2017.
- [3] Hüttermann, Michael. "Beginning devops for developers." In DevOps for Developers, pp. 3-13. Apress, Berkeley, CA, 2012.
- [4] Hüttermann, Michael. "Infrastructure as code." In DevOps for Developers, pp. 135-156. Apress, Berkeley, CA, 2012.
- [5] Ebert, Christof, Gorka Gallardo, Josune Hernantes, and Nicolas Serrano. "DevOps." *Ieee Software* 33, no. 3 (2016): 94-100.
- [6] Strode, Diane E., Sid L. Huff, and Alexei Tretiakov. "The impact of organizational culture on agile method use." In 2009 42nd Hawaii International

- Conference on System Sciences, pp. 1-9. IEEE, 2009.
- [7] Nerur, Sridhar, RadhaKanta Mahapatra, and George Mangalaraj. "Challenges of migrating to agile methodologies." *Communications of the ACM* 48, no. 5 (2005): 72-78.
 - [8] Hüttermann, Michael. "Introducing DevOps." In *DevOps for Developers*, pp. 15-31. Apress, Berkeley, CA, 2012.
 - [9] Rodríguez, Christian Adrián, Lía Hebe Molinari, and Francisco Javier Díaz. "The hard way to virtual machine administration: towards DevOps." *Journal of Computer Science & Technology* 13 (2013).
 - [10] Hemon, Aymeric, Barbara Lyonnet, Frantz Rowe, and Brian Fitzgerald. "From agile to DevOps: Smart skills and collaborations." *Information Systems Frontiers* 22, no. 4 (2020): 927-945.
 - [11] Jabbari, Ramtin, Nauman bin Ali, Kai Petersen, and Binish Tanveer. "What is DevOps? A systematic mapping study on definitions and practices." In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pp. 1-11. 2016.
 - [12] Mohamed, Samer I. "DevOps shifting software engineering strategy-value based perspective." *International Journal of Computer Engineering* 17, no. 2 (2015): 51-57.
 - [13] Bass, Len. "The software architect and DevOps." *IEEE Software* 35, no. 1 (2017): 8-10.
 - [14] Schaefer, Andreas, Marc Reichenbach, and Dietmar Fey. "Continuous integration and automation for DevOps." In *IAENG Transactions on Engineering Technologies*, pp. 345-358. Springer, Dordrecht, 2013.
 - [15] Rahman, Akond Ashfaqur, Eric Helms, Laurie Williams, and Chris Parnin. "Synthesizing continuous deployment practices used in software development." In *2015 Agile Conference*, pp. 1-10. IEEE, 2015.
 - [16] Mohammad, Sikender Mohsienuddin. "DevOps Automation Advances IT Sectors with the Strategy of Release Management." *International Journal of Computer Trends and Technology (IJCTT)–Volume* 67 (2019).
 - [17] Edwards, D. "Introducing DevOps to the Traditional Enterprise." (2014).
 - [18] Chen, Lianping. "Continuous delivery: Overcoming adoption challenges." *Journal of Systems and Software* 128 (2017): 72-86.
 - [19] Riungu-Kalliosaari, Leah, Simo Mäkinen, Lucy Ellen Lwakatare, Juha Tiihonen, and Tomi Männistö. "DevOps adoption benefits and challenges in practice: a case study." In *International Conference on Product-Focused Software Process Improvement*, pp. 590-597. Springer, Cham, 2016.
 - [20] Bruno, Dario, Thomas Fritz, Sharon Keidar-Barner, Philipp Leitner, Francesco Longo, Clarissa Marquazan, Andreas Metzger et al. "CloudWave: Where adaptive cloud management meets DevOps." In *2014 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1-6. IEEE, 2014.
 - [21] Paez, Nicolás. "Versioning Strategy for DevOps Implementations." In *2018 Congreso Argentino de Ciencias de La Informática y Desarrollos de Investigación (CACIDI)*, pp. 1-6. IEEE, 2018.
 - [22] Stillwell, Mark, and Jose GF Coutinho. "A DevOps approach to integration of software components in an EU research project." In *Proceedings of the 1st International Workshop on Quality-Aware DevOps*, pp. 1-6. 2015.
 - [23] Pérez, Juan F., Weikun Wang, and Giuliano Casale. "Towards a devops approach for software quality engineering." In *Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development*, pp. 5-10. 2015.
 - [24] Borgenholt, Gaute, Kyrre Begnum, and Paal Engelstad. "Audition: a DevOps-oriented service optimization and testing framework for cloud environments." (2013).
 - [25] Syed, Madiha H., and Eduardo B. Fernandez. "Cloud ecosystems support for Internet of Things and DevOps using patterns." In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 301-304. IEEE, 2016.
 - [26] Ivanov, Vitalii, and Kari Smolander. "Implementation of a DevOps pipeline for serverless applications." In *International Conference on Product-Focused Software Process Improvement*, pp. 48-64. Springer, Cham, 2018.
 - [27] Hosono, Shigeru. "A DevOps framework to shorten delivery time for cloud applications." *International Journal of Computational Science and Engineering* 7, no. 4 (2012): 329-344.
 - [28] van der Houven MSc, Wouter. "Security principles for DevOps and cloud."
 - [29] Yasar, Hasan, and Kiriakos Kontostathis. "Where to integrate security practices on DevOps platform." *International Journal of Secure Software Engineering (IJSSE)* 7, no. 4 (2016): 39-50.
 - [30] Wettinger, Johannes, Uwe Breitenbücher, Oliver Kopp, and Frank Leymann. "Streamlining DevOps automation for Cloud applications using TOSCA as standardized metamodel." *Future Generation Computer Systems* 56 (2016): 317-332.
 - [31] Kumar, Rakesh, and Rinkaj Goyal. "Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)." *Computers & Security* 97 (2020): 101967.
 - [32] Ur Rahman, Akond Ashfaqur, and Laurie Williams. "Security practices in DevOps." In *Proceedings of the Symposium and Bootcamp on the Science of Security*, pp. 109-111. 2016.
 - [33] Rios, Erkuden, Eider Iturbe, Wissam Mallouli, and Massimiliano Rak. "Dynamic security assurance in multi-cloud DevOps." In *2017 IEEE Conference on Communications and Network Security (CNS)*, pp. 467-475. IEEE, 2017.
 - [34] Arulkumar, V., and R. Lathamaju. "Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method." *Procedia Computer Science* 165 (2019): 399-405.
 - [35] Bahadori, Kiyana, and Tullio Vardanega. "DevOps meets dynamic orchestration." In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pp. 142-154. Springer, Cham, 2018.
 - [36] Laukkanen, Teemu, Kati Kuusinen, and Tommi Mikkonen. "Regulated software meets DevOps." *Information and Software Technology* 97 (2018): 176-178.

- [37] Morris, Dave, S. Voutsinas, Nigel C. Hambly, and Robert G. Mann. "Use of Docker for deployment and testing of astronomy software." *Astronomy and computing* 20 (2017): 105-119.
- [38] Sandoval, Robert. "A case study in enabling DevOps using Docker." PhD diss., 2015.
- [39] Doukoure, Gaoussou Abdel Kader, and Ernest Mnkandla. "Facilitating the Management of Agile and DevOps Activities: Implementation of a Data Consolidator." In *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1-6. IEEE, 2018.
- [40] Perez-Palacin, Diego, Youssef Ridene, and José Merseguer. "Quality assessment in DevOps: automated analysis of a tax fraud detection system." In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*, pp. 133-138. 2017.
- [41] Spinellis, Diomidis. "Don't install software by hand." *IEEE software* 29, no. 4 (2012): 86-87.
- [42] Orr, Anthony. "Maximize the synergies between ITIL and DevOps." *BMC Software* (2012).
- [43] Bucena, Ineta, and Marite Kirikova. "Simplifying the DevOps Adoption Process." In *BIR Workshops*, pp. 1-15. 2017.
- [44] Khan, Muhammad Owais, Awais Khan Jumani, and Waqas Ahmed Farhan. "Fast Delivery, Continuously Build, Testing and Deployment with DevOps Pipeline Techniques on Cloud." *INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY* 13, no. 05 (2020): 552-575.
- [45] Rafi, Saima, Wu Yu, Muhammad Azeem Akbar, Ahmed Alsanad, and Abdu Gumaei. "Multicriteria based decision making of DevOps data quality assessment challenges using fuzzy TOPSIS." *IEEE Access* 8 (2020): 46958-46980.
- [46] Luz, Welder Pinheiro, Gustavo Pinto, and Rodrigo Bonifácio. "Adopting DevOps in the real world: A theory, a model, and a case study." *Journal of Systems and Software* 157 (2019): 110384.
- [47] Heinrich, Robert, Reiner Jung, Christian Zirkelbach, Wilhelm Hasselbring, and Ralf Reussner. "An Architectural Model-Based Approach to Quality-Aware DevOps in Cloud Applicationsc." In *Software Architecture for Big Data and the Cloud*, pp. 69-89. Morgan Kaufmann, 2017.
- [48] Mikkonen, Tommi, Casper Lassenius, Tomi Männistö, Markku Oivo, and Janne Järvinen. "Continuous and collaborative technology transfer: Software engineering research with real-time industry impact." *Information and Software Technology* 95 (2018): 34-45.
- [49] Solouki, Soha. "Knowledge Management Practices in DevOps." PhD diss., Université d'Ottawa/University of Ottawa, 2020.
- [50] McCarthy, Matthew A., Lorraine M. Herger, Shakil M. Khan, and Brian M. Belgodere. "Composable DevOps: automated ontology-based DevOps maturity analysis." In *2015 IEEE International Conference on Services Computing*, pp. 600-607. IEEE, 2015.
- [51] Airaj, Mohammed. "Enable cloud DevOps approach for industry and higher education." *Concurrency and Computation: Practice and Experience* 29, no. 5 (2017): e3937.
- [52] Morales, Jose Andre, Hasan Yasar, and Aaron Volkman. "Implementing DevOps practices in highly regulated environments." In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, pp. 1-9. 2018.
- [53] Wiedemann, Anna, and Manuel Wiese. "Are you ready for Devops? Required skill set for Devops teams." (2018).