

A COMPREHENSIVE STUDY OF DEVOPS USAGE IN PRACTICE

V. Agaphi Bob

Assistant Professors, Ramachandra College of Engineering,

Eluru, Andhra Pradesh, India.

vagaphib.932@gmail.com

Abstract

Organizations are introducing agile and lean software development techniques in operations to increase the pace of their software development process and to improve the quality of their software. They use the term DevOps, a portmanteau of development and operations, as an umbrella term to describe their efforts. In this paper we describe the ways in which organizations implement DevOps and the outcomes they experience. We first summarize the results of a Systematic Literature Review that we performed to discover what researchers have written about DevOps. DevOps is an emerging practice to be followed in Software Development life cycle. The name DevOps indicates that it's an integration of 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.

1. Introduction:

The practices followed during the software development lifecycle plays an important role. In the conventional development lifecycle, different teams will play their role at their own level. Separate teams make the product life cycle lengthier and also the communication between the teams poor. 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. 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 (Michael Hüttermann 2012). DevOps cannot be implemented at one stretch, it needs to be implemented step by step or iteratively. DevOps was divided into 4 areas (Michael Hüttermann 2012) and it is shown in figure (Christof Ebert et al., 2016). 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.

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 (Strode et al., 2009). The agile is the evolutionary development model (Nerur et al., 2005) 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 (Michael Hüttermann 2012). It was hard to maintain the software and to update them whenever necessary (Rodríguez et al., 2013). The agile processes are feature and people-centric approaches and so it was a challenge to change from traditional process-centric approach (Nerur et al., 2005). 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 (Hemon et al., 2020). This gives rise to the new methodology DevOps, which combines the Development and Operations team to enhance communication and frequency.



DevOps

DevOps is a set of procedures which combines the process of Development and Operations (Christof Ebert et al., 2016). 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 (Christof Ebert et al., 2016). The DevOps has four

Dimensions (Lwakatare et al., 2015), they are

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

DevOps is the extension of the agile method of software development (Jabbari et al., 2016) . DevOps focuses on the continuous delivery of the software along with continuous integration (Jabbari et al., 2016 and Manish Virmani, 2015). 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 (Samer Mohamed 2015).

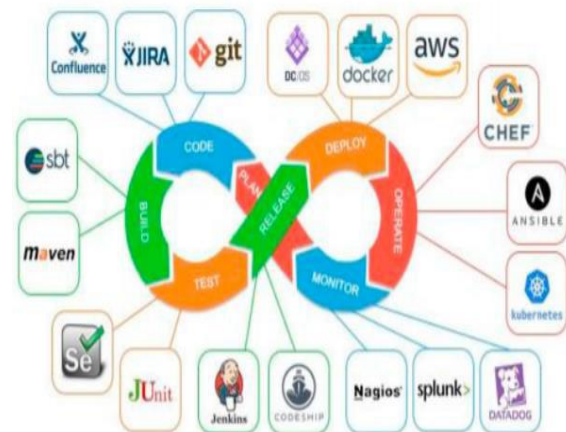
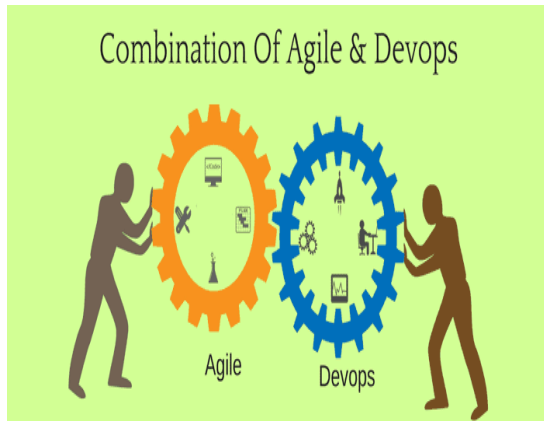


Fig: tools used in DevOps

3. DevOps and Agile Relationship

Before Agile, Companies used to follow the traditional approach, i.e. Waterfall model for software development. To overcome the gaps introduced by the Waterfall model, companies have adapted Agile Methodology.



While DevOps addresses the communication gap between the development team and operation team, Agile addresses the communication gap between customer requirements and the development team.

Agile Practices that Align with DevOps

- Agile's iterative approach ensures **Iterative Development and Continuous Feedback**
- continuous feedback, which is crucial for DevOps processes.
- Regular iterations allow for constant improvements and adjustments based on real-time data.

Collaboration and Communication

- Both Agile and DevOps emphasize the importance of collaboration and communication.
- Agile's cross-functional teams and DevOps' integration of development and operations break down silos and foster a collaborative environment.

Flexibility and Adaptability

- Agile's focus on adaptability aligns well with DevOps' need for flexible processes.

- The ability to respond quickly to changes is a key strength of both methodologies.

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 (Leah et al., 2016, Khan et al., 2020). The challenges involved can be categorised as lack of awareness, support, technological feasibility and adapting to the change (Bucena and Marite 2017). 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 (Chen et al., 2017). The infrastructure also plays an important role in DevOps adoption and it should be compatible and lightweight (Khan et al., 2020). The challenges identified using the Fuzzy TOPSIS approach in (Rafi et al., 2020) were Data heterogeneity, data integration, error and inconsistent data, a misspelling in data entry, missing information, traceability of data, Data harmonization, visualisation 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 (Michael Hüttermann 2012). Some of the obstacles in software development for a highly regulated environment are 1) the operator doesn't know about the code 2) all artefacts

are stored in a single isolated container 3) project must be approved before commencement 4) Poor collaboration (Morales et al., 2018).

5. Proposed Models

A three-step model for DevOps adoption was proposed in (Luz et al., 2019). 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 (Luz et al., 2019). The iObserve approach used to tackle the challenges in DevOps adoption was discussed in (Heinrich et al., 2017). The iObserve approach follows MAPE (Monitor, Analyse, Plan, Execute) Control loop. The framework of iObserve approach.

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 (Leah et al., 2016). The set of metrics used to assess and evaluate the DevOps practices were discussed in (Leah et al., 2016). The proposed organisational 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 (Perez-Palacin et al., 2017), the Simtool was used to evaluate the metrics. The Fuzzy TOPSIS approach was used for the quality assessment in DevOps (Rafi et al., 2020).

7. Future works on DevOps:

By changing the configuration of the existing model, the upcoming model can be developed as multi-dimensional (Arulkumar and Lathamanju 2019). The DevOps team will possess desirable qualities or skills, full-stack development, analysis, functional, decision-making, social, testing, and advisory skills (Wiedemann et al., 2018). In the future, most of the IT firms will adopt the continuous deployment strategy (Rahman et al., 2015). The new metrics will be found to evaluate the quality of the DevOps development cycle (Leah et al., 2016). Tools like agile consolidator aimed at improving the quality of the process should be built (Doukoure et al., 2018). The auto-scaling policies in the combination of both container and node level will be developed (Kiyana Bahadori and Tullio Vardanega 2018). Some future works that can be done on working with Docker containers are Networking, Eclipse Plugin, Port management and Clustering capabilities (Robert Sandoval 2016). External clients are eager to collaborate with focused groups inside the solution providing enterprises to develop mixed solution assets in an agile fashion (McCarthy and Lorraine 2015).

8. Conclusion:

In this paper, we have discussed the building blocks of DevOps, challenges in adopting DevOps and the future work. DevOps is an emerging optimal practice that should be followed in a software development life cycle to increase releases, reliability, faster updates, and 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 by small increments. The 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, and security and in collecting feedback. Many frame-work 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. *DevOps: A Historical Review and Future Works* by Mayank Gokarna, Raju Singh
2. Ebert, Christof, Gorka Gallardo, Josune Hernantes, and Nicolas Serrano. "DevOps." *Ieee Software* 33, no. 3 (2016): 94-100.
3. Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N., 2012, "A decade of agile methodologies: Towards
4. Drury, M., Conboy, K., and Power, K., 2012, "Obstacles to decision making in Agile software development teams", *Journal of Systems and Software*, 85(6), pp. 1239- 1254.
5. 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.
6. Bucena, Ineta, and Marite Kirikova. "Simplifying the DevOps Adoption Process." In *BIR Workshops*, pp. 1-15. 2017.
7. Gill, Asif Qumer, Abhishek Loumish, Isha Riyat, and Sungyoun Han. "DevOps for information management systems." *VINE Journal of Information and Knowledge Management Systems* (2018).
8. Rahman, Akond Ashfaque Ur, Eric Helms, Laurie Williams, and Chris Parnin. "Synthesizing continuous deployment practices used in software development." In *2015 Agile Conference*, pp. 1-10. IEEE, 2015.
9. 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.
10. 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.
11. 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.
12. Qumer, Asif, Brian Henderson-sellers, and Tom Mcbride. "Agile adoption and improvement model." (2007).
13. 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.