

DevOps: Breakdown the Development-Operations Hurdle Barrier

Mrs. Vishakha Nilesh Pawar¹, Mrs. Gayatri Rakesh Jagtap², Ms. Preeti Balu Kudal³

Lecturer, Guru Gobind Singh Polytechnic, Nashik, Maharashtra, India^{1,3}

HOD, Guru Gobind Singh Polytechnic, Nashik, Maharashtra, India²

vishakhaate@gmail.com¹, gayatri.jagtap@ggsf.edu.in², preeti.kudal@ggsf.edu.in³

Abstract: *There are numerous conflicts among development team and operations team when they are developing and delivering important software to clientele in software development process. DevOps, as a significant rising concept, is proposed to surmount the conflict between development team and operations team. Many companies and organizations have a tendency to adapt DevOps. The DevOps drive commenced around 2007 when the software development and IT operations societies elevated apprehensions about the traditional software development model, where developers who engraved code functioned apart from operations who deployed and maintained the code. The term DevOps, a combination of the arguments development and operations, imitates the process of assimilating these disciplines into one, continuous process. DevOps as a relatively rising concept have requirement of understanding the possible challenges and mitigation strategies.*

Keywords: DevOps, Challenge, Systematic Literature review

I. INTRODUCTION

In many software organizations, the departments are separately organized for development, IT operations and Quality Assurance. Most of the time conflicts occur between development teams and operations team when they are developing and delivering various important software to customers during their regular routine work [1]. Stability and security are important factor in operations and same is implied by developers do not change products frequently, whereas developers show their more interest in deploying a new features or changes to clientele [1] [2] [3]. Operations team members are not ready to accept new versions releasing too quickly [4]. These conflicts can hamper software development [4]. DevOps word is a formed by blending two terms together that is development and operations, which are used to conduit the gaps between development team and operations team by adjusting the incentives and sharing approaches for the whole development processes [1]. DevOps is a blend of patterns to perk up the collaboration and communication between developers and operations team members to decipher critical issues during software development [8]. The important serious issues are like fear of changes and precarious deployment [8]. DevOps mitigates the gap between developers, operations and end users, and allows earlier issue detection [7]. DevOps could accept the all routine challenges in software deployment and delivery and provides the various steps to deal with them [8].

To understand the scope of DevOps properly, it is very crucial to be aware of aspects of software engineering that are not DevOps. There is no DevOps methodology or process [7] and also it is not a new department [8]. DevOps and Agile are similar, but are somewhat different in some of the important aspects [5]. Agile represents thinking of changes; rather DevOps implements organizational changes [5]. Comparing DevOps with Agile, DevOps is more like a conceptual framework [7]. Due to the development of DevOps, business companies start to apply DevOps during development process. In other side, DevOps involves various activities and aspects as follows [8]:

- Automation: For DevOps, automation is necessary to get quick feedback.
- Culture: In DevOps, people are more important than the process and tools since the software is made by personal and used by personal.
- Measurement: DevOps finds a particular measure path. Quality and shared incentive are critical.

- Sharing: Create an ethnicity for people to share tools, ideas, and opinions.

1.1 What is DevOps?

DevOps is an idea, which has far not been regularly discussed in the academic literature [7]. DevOps is mainly based upon the ideas from agile development movements and supports rapid development and deployment cycles [19]. The definition for DevOps is not widely accepted. Olszewska and Waldén reflect on that DevOps is a software development technique that combines Quality Assurance with Operations in development practices [19]; S. Farroha and L. Farroha states that DevOps is a strategy that could be used in business to describe a enhanced work way between application development and infrastructure operations professionals [21]; DevOps are defined as an emerging paradigm which improves the collaboration and minimize split and barrier in two teams; development teams and operations teams by Wettinger, Andrikopoulos, and Leymannjust [22] [24] [25]; Stackpole cogitates that DevOps is further of a culture shift than a full-blown development methodology, with the assistance of a suite of automated tools, put emphasis on early association between the operations and development teams [14]; according to [26], DevOps is measured as a set of methods for make things easier and fit in the development and operations of software development process; Dyck, Penners, and Lichter recommend that DevOps should be an structural approach, which “pressures empathy and cross-functional collaboration inside and between teams –specifically development and IT operations –in software development organizations, in order to function resilient systems and quicken delivery of modifications.” [27] The descriptions of DevOps stated above are just chunk of definitions from the papers that are reviewed. The main Noticeable part after reviewing is that there is no uniform definition of the nature of DevOps.

In order for the writers to spread a consistent accepting of DevOps during the research process, the writers framed the definition of DevOps as follows[61]: DevOps should be a manner of work, by means of a sequence of tools to intensification the communication and collaboration between development and operations teams to diminish the clashes between the two teams and increase the development efficiency and quality. According to IBM Cloud [28], DevOps permit developers, testers and operations work in alliance using shared DevOps Sequence of tools, and supports persistently deliver software by permitting cooperative testing and continuous monitoring during the development, environment. Tools play a substantial role in DevOps, which could simplify version management, infrastructure configuration, instrumentation, monitoring, containerization, virtualization, and automation. The DevOps community put together open source tools with Nomadic (for creating and configuring virtual development environments) that use existing configuration management tools such as Puppet and Chef from 2011.

1.2 Benefits of DevOps

The complete planned objective of DevOps is to realize the best return on investment, at an equivalent time, make sure the quality of software and satisfy the requirements of consumers [21]. DevOps tries to deliver a continuous channel to support continuous delivery of software in order to empower fast and frequent releases [24], automatically testing cycles [33]. DevOps also facilitate quick responses to vary requirements from customers [24]. With DevOps, developers and operations could work together by integrating all organizational systems, simplifying testing and quality assurance [31], and smoothen out and conduit the gap between development and operations [19] [6]. DevOps opens the chances of eliminating the riven of organizational and cultural challenges [30], and addresses the value for defect identification during the first stages [32]. within the DevOps environment, bugs within the code are immediately corrected early within the software development lifecycle due to the continued deployment of software builds [32].

1.3 Ambiguity of DevOps

In addition to benefits of DevOps, there are many experiments of practicing DevOps and this is often why we do the research for this paper. For instance, there is an deceptive problem that the definition of DevOps is ambiguous, which can make misunderstanding when applying it during a real-world development process. In this paper, we classified the experiments into several categories supported the results of Literature Review, and there'll be an in depth explanation within the following sections.

II. RELATED WORK

DevOps is an evolving concept, in runthrough, we will meet many unfamiliar challenges. At present, there are limited research on the challenges of DevOps, so there are still many challenges when relating DevOps. In this article, we analysis relevant articles, recognize existing challenges, and practice a survey to explore the expediency of these challenges in industry. As can be seen from the associated articles, DevOps has transformed the way people work in the previous era, so the experiments of culture and the experiments of personnel are critical. The main aim of this paper is to Find out the challenges of DevOps and mitigation approaches, through our research to help people who use DevOps to identify and diminish the risk of the challenges during the development process. Some organizations do not have sufficient skilled staffs to implement DevOps entirely was introduced by B. S. Farroha and D.L. Farroha [21]. Also, when there are no robust security engineers in the team, security and compliance be likely to be damaged [21]. A new language called DevOpSlang which is used to thorough DevOps in the organization was introduced by Wettinger et al. [24]. There is no precise DevOps model that can be pragmatic to all companies was stated by Erich et al. [40]. The shared challenge is still cultural change rather than technology is a thought of Taft and Darryl [45]. The commanding open problems in software evolution and pronounces how to document interpretation centric DevOps in the associations was outlined by Gottesheim (11). He also introduces that platoon-to- platoon prognosticating and blame can be circumvented by depicting and participating performance criteria across brigades when looked with problems. Shahin (41) proposed that DevOps and nonstop deployment can be challenging for software engineers, so the operation should bere-architected to support a variety of DevOps practices. McCarthy et al. (42) introduced a frame to gradationally ameliorate being DevOps practices into further cohesive and cooperative practices and to measure the value of collaboration. Olszwska and Walden (43) introduced how to formalize modeling in DevOps, and how to ensurequality informal modeling in DevOps. Lwakatare et al. (44) defined the main aspects of DevOps are collaboration, robotization, dimension, and monitoring and also developed a frame to understand how DevOps works. Wettinger et al. (22) presented a holistic approach to landing DevOps knowledge to a knowledge base and manage it. Lwakatare and Dyck et al. (6) introduced there's a lack of common understanding of what DevOps constitutes in academia and the interpreters' communities.

Fredrickson (33) allowed the primary challenge is not specialized challenges, but communication challenges. Smeds, Nybom, and Porres (6) proposed that geographical distribution could produce challenges, for illustration, as communication cannot be done in person and as reaching people might be delicate due to different time zones.

Preimesberger and Chris said (31), conforming and balancing the pretensions and impulses must be used to address artistic challenges. Communicating and celebrating the success of DevOps in the development processis a critical strategy for reducing fear and structure business cases continuously. Wettinger et al. (18) banded that DevOps vestiges are generally bound to given tools, which make it grueling to exercise different kinds of miscellaneous vestiges in combination with others. Wahaballa et al. (29) defined a abstract deficiency problem which is caused by the collaboration between elaboration and operations brigades. At the same time, they handed a unified DevOps model (UDOM) to overcome this problem.

Because DevOps is a new conception, there aren't numerous high- quality studies on DevOps. At the same time, there's short literature exclusively concentrate on exploration DevOps challenges and mitigation strategies. Amaradri and Nutalapati (39) proposed the benefits, challenges, and mitigation strategies of DevOps, but the challenges arenot distributed. The authorstry to collect all the DevOps challenges and mitigations that live in the literature and in practice and examine them in detail.

III. DEVOPS PRACTICES

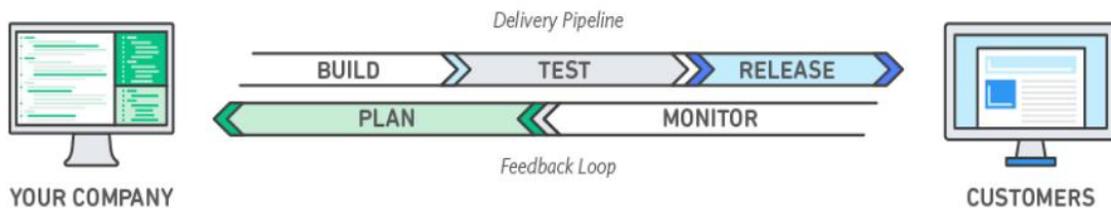
DevOps is a set of applies that mechanizes the processes between software development and IT teams, in order that they can build, test, and release software quicker and further reliably. The concept of DevOps is originated on building an ethos of association between teams that historically functioned in relative siloes.

3.1 DevOps Model Defined

DevOps is the amalgamation of cultural philosophies, applies, and tools that upsurges an organization’s ability to distribute applications and services at great rapidity: evolving and improving products at a quicker pace than organizations using traditional software development and infrastructure management processes. This rapidity empowers organizations to well serve their customers and compete more effectively in the market.

3.2 How DevOps Works

Under a DevOps model, development and operations teams are no elongated. Occasionally, these two teams are combined into a single team where the engineers exertion across the entire application lifecycle, from development and test to deployment to operations, and change a range of skills not limited to a single function. In some DevOps models, quality assurance and security teams might also develop more firmly incorporated with development and operations and throughout the application lifecycle. When security is the emphasis of everyone on a DevOps team, this is occasionally referred to as DevSecOps. These teams use performs to automate procedures that factually have been manual and slow. They use a technology stack and tooling which help them operate and evolve applications quickly and reliably. These tools also help engineers self sufficiently achieve tasks (for example, deploying code or provisioning infrastructure) that usually would have required assistance from other teams, and this further upsurges a team’s rapidity.



3.3 Benefits of DevOps



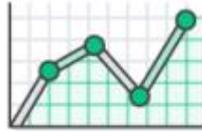
Speed: Transfer at high rapidity so you can modernize for customers quicker, adapt to fluctuating markets better, and grow further effectual at powerful business results. The DevOps model allows your developers and operations teams to achieve these results. For example, micro services and continuous delivery let teams take title of services and then release updates to them quicker.



Fast Delivery: Increment the recurrence and speed of deliveries so you can advance and further develop your item quicker. The speedier you can deliver new elements and fix messes with, the quicker you can react to your clients' requirements and assemble upper hand. Persistent joining and constant conveyance are rehearses that robotize the product discharge process, from work to send.



Unwavering Quality: Guarantee the nature of utilization updates and foundation changes so you can dependably convey at a more quick speed while keeping a positive encounter for end clients. Use rehearses like nonstop combination and ceaseless conveyance to test that each change is useful and safe. Checking and logging rehearses assist you with remaining educated regarding execution progressively.



Scale: Work and deal with your framework and advancement processes at scale. Computerization and consistency assist you with overseeing perplexing or changing frameworks productively and with decreased danger. For instance, foundation as code assists you with dealing with your turn of events, testing, and creation conditions in a repeatable and more proficient way.



Further Developed Collaboration: Fabricate more powerful groups under a DevOps social model, which underscores esteems like proprietorship and responsibility. Engineers and tasks groups team up intently, share numerous obligations, and join their work processes. This lessens shortcomings and saves time (for example decreased handover periods among engineers and tasks, composing code that considers the climate in which it is run).

Security: Move rapidly while holding control and saving consistence. You can embrace a DevOps model without forfeiting security by utilizing mechanized consistence approaches, fine-grained controls, and arrangement the board strategies. For instance, involving framework as code and strategy as code, you can characterize and afterward track consistence at scale.

Why DevOps Matters : Programming and the Internet have changed the world and its businesses, from shopping to diversion to banking. Programming never again only backings a business; rather it turns into a fundamental part of all aspects of a business. Organizations communicate with their clients through programming conveyed as online administrations or applications and on a wide range of gadgets. They additionally use programming to increment functional efficiencies by changing all aspects of the worth chain, like strategies, correspondences, and activities. Likewise that actual merchandise organizations changed how they configuration, construct, and convey items utilizing modern computerization all through the twentieth century, organizations in this day and age should change how they assemble and convey programming.

DevOps Practices

The following are DevOps best practices:

- Continuous Integration
- Continuous Delivery
- Microservices
- Infrastructure as Code
- Monitoring and Logging
- Communication and Collaboration

Below you can learn more about each particular practice.



Continuous Integration

Persistent coordination is a product advancement practice where designers consistently combine their code changes into a focal archive, after which robotized assembles and tests are run. The vital objectives of ceaseless joining are to find and address messes with faster, further develop programming quality, and lessen the time it takes to approve and deliver new programming refreshes.



Continuous Delivery

Consistent conveyance is a product improvement practice where code changes are naturally fabricated, tried, and ready for a delivery to creation. It develops ceaseless coordination by sending all code changes to a testing climate and additionally a creation climate after the form stage. At the point when consistent conveyance is carried out appropriately, engineers will forever have an arrangement prepared form antiquity that has finished through a normalized assessment process.



Micro Services

The micro services engineering is a plan way to deal with construct a solitary application as a bunch of little administrations. Each help runs in its own interaction and speaks with different administrations through an obvious connection point utilizing a lightweight instrument, ordinarily a HTTP-based application programming connection point (API). Micro services are worked around business capacities; each assistance is checked to a solitary reason. You can utilize various systems or programming dialects to compose micro services and send them autonomously, as a solitary help, or collectively of administrations.



Infrastructure as Code

Infrastructure as code is a training where framework is provisioned and overseen utilizing code and programming advancement procedures, for example, adaptation control and persistent joining. The cloud's API-driven model empowers designers and framework chairmen to interface with foundation automatically, and at scale, rather than expecting to physically set up and arrange assets. Accordingly, specialists can interact with foundation utilizing code-based apparatuses and treat framework in a way like how they treat application code. Since they are characterized by code, foundation and waiters can rapidly be conveyed utilizing normalized designs, refreshed with the most recent patches and forms, or copied in repeatable ways.

Configuration Management

Developers and system administrators use code to automate operating system and host configuration, operational tasks, and more. The use of code makes configuration changes repeatable and standardized. It frees developers and systems administrators from manually configuring operating systems, system applications, or server software.

DevOps Tools

The DevOps model depends in compelling tooling to help groups quickly and dependably convey and advance for their clients. These apparatuses mechanize manual undertakings, assist groups with overseeing complex conditions at scale, and keep engineers in charge of the great speed that is empowered by DevOps. AWS offers types of assistance that are intended for DevOps and that are constructed first for use with the AWS cloud. These administrations assist you with utilizing the DevOps rehearses depicted previously.

REFERENCES

- [1]. Michael Hüttermann, "Introduction," in DevOps for Developers, 2012 edition, Berkeley, USA, Apress, September 2012.
- [2]. M. Loukides, What is DevOps?, 1 edition. O'Reilly Media, 2012.
- [3]. Floris, Erich; Chintan, Amrit; Maya, Daneva (2014-12-10). "A Mapping Study on Cooperation between Information System Development and Operations"
- [4]. Michael Hüttermann, "Introducing DevOps," in DevOps for Developers, 2012 edition, Berkeley, USA, Apress, September 2012, pp. 23–23.
- [5]. Ambler, Scott W. (12 February 2014). "We need more Agile IT Now!". Dr. Dobb's The world of software Development (San Francisco: UBM).
- [6]. J. Smets, K. Nybom, and I. Porres, "DevOps: A Definition and Perceived Adoption Impediments," in Agile Processes, in Software Engineering, and Extreme Programming, C. Lassenius, T. Dingsøyr, and M. Paasivaara, Eds. Springer International Publishing, 2015, pp. 166–177.
- [7]. F. Erich, C. Amrit, and M. Daneva, "Report: Devops literature review," University of Twente, Tech. Rep., 2014.
- [8]. Michael Hüttermann, "Beginning DevOps for Developers," in DevOps for Developers, 2012 edition, Berkeley, USA, Apress, September 2012, pp. 4–13.
- [9]. C. Wohlin and A. Aurum, "Towards a decisionmaking structure for selecting a research design in empirical software engineering," Empirical Software Engineering, vol. 20, no. 6, pp. 1427–1455, 2015.
- [10]. B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," in Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.
- [11]. W. Gottesheim, "Challenges, Benefits and Best Practices of Performance Focused DevOps," in Proceedings of the 4th International Workshop on Large-Scale Testing, New York, NY, USA, 2015, pp. 3–3.
- [12]. Lisa M. Given, SAGE Encyclopedia of Qualitative Research Methods, USA: SAGE, 2008.
- [13]. "Global Study by CA Technologies Finds DevOps is a Key Contributor to the Bottom Line -CA Technologies." [Online]. Available: <http://www.ca.com/us/company/newsroom/pressreleases/2015/global-study-by-ca-technologiesfinds-devops-is-a-key-contributor-to-the-bottomline.html>. [Accessed: 02-Jan-2017].
- [14]. B. Stackpole, "5 DISRUPTORS to Keep on Your Radar. (cover story)," Computerworld Digital Magazine, pp. 19–22, Dec. 2015.
- [15]. A. Pinsonneault and K. Kraemer, "Survey research methodology in management information systems: an assessment," Journal of management information systems, vol. 10, no. 2, pp. 75–105, 1993.
- [16]. C. B. Seaman, "Qualitative methods iinn empirical studies of software engineering," IEEE Transactions on Software Engineering, vol. 25, no. 4, pp. 557–572, Jul. 1999.
- [17]. C. Bennett, S. Khangura, J. C. Brehaut, I. D. Graham, D. Moher, B. K. Potter, and J. M. Grimshaw, "Reporting guidelines for survey research: an analysis of published guidance and reporting practices," PLoS Med, vol. 8, no. 8, p. e1001069, 2011.
- [18]. I. Weber, S. Nepal, and L. Zhu, "Developing Dependable and Secure Cloud Applications," IEEE Internet Computing, vol. 20, no. 3, pp. 74–79, 2016.
- [19]. M. Olszewska and M. Waldén, "DevOps meets formal modelling in high-criticality complex systems," in Proceedings of the 1st International Workshop on Quality-Aware DevOps, 2015, pp. 7–12.

- [20]. D. Edwards. The History Of DevOps. <http://itrevolution.com/the-history-of-devops/>, September 17, 2012. [Online; accessed: 11-9-2016].
- [21]. B. S. Farroha and D. L. Farroha, "A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment," in 2014 IEEE Military Communications Conference, 2014, pp. 288–293.
- [22]. J. Wettinger, V. Andrikopoulos, and F. Leymann, "Automated capturing and systematic usage of devops knowledge for cloud applications," in Cloud Engineering (IC2E), 2015 IEEE International Conference on, 2015, pp. 60–65.