International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 4 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195365

DevOps - A collaboration Of Developing & Deploying Software

Parinita Agarwal
Radha Govind Group of Institutions, Meerut, Uttar Pradesh, India

ABSTRACT

DevOps is the fuse of tasks performed by a company's application development and systems operations teams. The team DevOps is being used in distinct ways. DevOps is a major innovation in Information System development. In its most broad definition, DevOps is an operational ideology that stimulate better communication between development and operations as more elements of operations become programmable. The DevOps approach seeks to meld application development and deployment into a more streamlined process that aligns development quality assurance (QA) and operations team efforts. This approach also shifts some of the operation team's responsibilities back to the development team in order to facilitate continuous development, continuous integration, continuous delivery and continuous monitoring processes. The necessity for tearing down the silos between development and operations has been expedited by the need to release code faster and more often in order to help the organization respond in a more agile manner to changing business requirements. Other drivers for breaking down the silos include the increasing use of cloud computing and advances in software defined infrastructures, microservices, containers and automation.

Keywords : Devops, Software Engineering, Agile

I. INTRODUCTION

DevOps (a clipped compound of "development" and "operations") is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.

DevOps can be seen as a business-driven software delivery approach, an approach that takes a new or enhanced business idea from design to development to testing and all the way to production, providing business value to customers in an efficient manner.

1. Devops

Patrick Debois, who's often called "the father of DevOps", coined the word "DevOps" in 2009. As the word depicts, it was formed by combining two words: "development" and "operations". DevOps is a collaborative way of developing and deploying software. DevOps (a portmanteau of development and operations) is a software development method that stresses communication, collaboration and integration between software developers and information technology (IT) operation professionals [1].

- DevOps is an approach based on agile and lean principles in which business owners, development, operations, and quality assurance team collaborate to deliver software in a continuous stable manner

- DevOps is an environment that promotes cross practicality, shared business tasks and belief
- DevOps is a movement that improves IT service delivery agility
- DevOps is a culture that promotes better working relationship within the company
- DevOps is a set of practices that provides rapid, reliable software delivery

2. DevOps Lifecycle

As DevOps is intended to be a cross-functional mode of working, rather than a single DevOps tool there are sets (or "toolchains") of multiple tools. Such DevOps tools are expected to fit into one or more of these categories, reflective of key aspects of the development and delivery process. DevOps is deep integration between development and operations. Understanding DevOps is not possible without knowing DevOps lifecycle.

II. METHODS AND MATERIAL

Continuous DevOps life-cycle:

1. Development

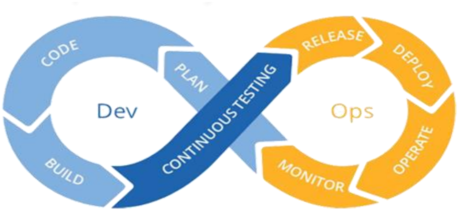
In this DevOps stage the development of software takes place constantly. In this phase, the entire development process is separated into small development cycles. This benefits DevOps team to speed up software development and delivery process. Continuous development encompasses coding then creating different forms of the code using SVN, Git and then wrapping the code into an executable folder to forward it to the quality analysts for testing the codes.

2. Testing

Testing is a process that coincides with the actual use of the application in the DevOps. Beta testers can produce results while still ensuring that the application can have its intended use in a live environment. The testing provides further information about different aspects of the application, which in turn, can again be sent to the development process to improve the application [3]. Quality Analysts use tools like Selenium, Junit, etc. to remove the bugs while testing the developed software and to ensure that there are no flaws in the functionality of the software.

3. Integration

The continuous integration is the process that add new functionality is integrated with the prevailing code, and testing takes place. Continuous development is only possible due to continuous integration and testing. The integration includes various steps, such as planning the various tests that are going to be carried out in the next phase, as well as understanding whether the developed code can produce the desired operations as required. Jenkins tool is used in the integration purpose that triggers any change made in the code automatically or even manually.



4. Deployment

The deployment process takes place continuously. It is performed in such a manner that any changes made any time in the code, should not affect the functioning of high traffic website. automation tools like the SaltStack, Puppet, Chef etc. plays an important role.

5. Monitoring

The operational phase in DevOps is actually the phase of monitoring, in this phase operation team will take care of the inappropriate system behavior or bugs which are found in production [2]. For this, the team uses Popular tools like Sensu, NewRelic, and Nagios.

4. Why DevOps is needed??

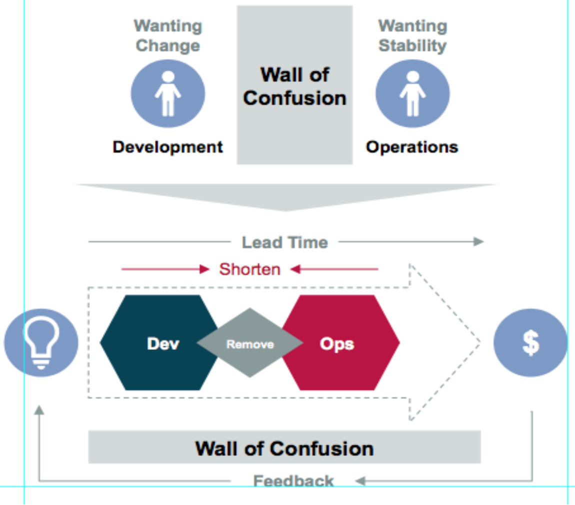
Traditionally what happens is that software developers write the codes for weeks and months and handover the code to the Quality Assurance team for testing which then handover the finale release to operation team for deployment. There is a lack of collaboration between all these phases i.e. development, testing, and deployment.

Developers write the code and hands over to the deployment team. Now it's up to deployment team to fix the problems that arise during deployment of code or hands over the code altogether back to the development team to fix the bugs. All this results in a slowing of software development process.

Developers always want to deliver changes as soon as possible. Operations want reliability and stability.

Lee Thomson describes this as a wall of confusion between development and operations [4]. This wall of confusion not only exists between the mindsets of the two teams but also with the tools they use. Development uses some tools and operation uses some other tools to perform the same stuff.

- The conflict of interest between the goals of those who create the software product and those who maintain it leads to poor relations,
- Development teams focus on building software that meets functional and quality requirements to create effective change whilst Operations teams focus on operability (e.g. availability, stability), service improvement and cost efficiency,
- Breaking barriers across the extended Development and Operations organization will require changes including reshuffle and relocation of several team members,



- With such changes one can expect resistance from the functional silos when trying to adopt a DevOps mindset for development and operations teams,
- Strong resistance can be expected especially from people who do not see the benefits that might be gained from it who are unwilling to change their attitude or shift their behavior,
- People who are not used to working in cohesive, cross functional teams will find it difficult to adjust to the new normal.
- Teams that are culturally not disciplined, which is very essential for Agile methodologies and DevOps, may not realize desired benefits,
- Development and Operations teams blame each other when something goes wrong,
- Communication challenges could arise between team members due to a lack of common terminology.

But under the umbrella of DevOps model, these three teams are no longer isolated from each other. Most of the times these three teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function. Security teams may also become more tightly integrated with development and operations and throughout the application lifecycle.

DevOps break down the walls between development and operations team, unifying development to operations for better, faster outcomes. It breaks the communication challenges. It helps the people work in cohesive manner. Neither the developers nor the operations team blame each other if anything goes wrong.

DevOps allow the adaptation of new technologies while developing a software, it helps the developers to adapt the changes without any resistance.

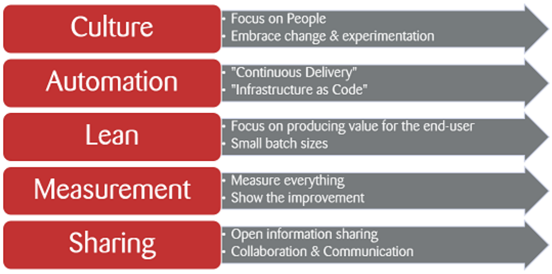
5. DevOps Principles

- Customer-Centric Action: Nowadays it is crucial to have short feedback loops with real customers and end-users. To be able to meet these customers' requirements, DevOps team must take customer-centric action for that they should constantly invest in products and services.
- End -To-End Responsibility: The DevOps team need to provide performance support until they become end-of-life. This enhances the level of responsibility and the quality of the product engineered.
- Continuous Improvement: DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the improvement of product or services offered.
- Automate everything: To adopt a continuous improvement culture with and to create an IT organization that receives instant feedback from end users or customers, many organizations have quite some waste to eliminate. Automation is a vital principle of DevOps process. This is not only for the software development but also for the entire infrastructure landscape.
- Work as one team: In the DevOps culture role of the designer, developer, and tester are already defined. All they needed to do is work as one team with complete collaboration.
- Monitor and test everything: It is very important for DevOps team to have a robust monitoring and testing procedures for achieving high quality and stable product and services with increase efficiency and capacity and with better velocity of delivery.

6. DevOps A Way of Thinking: CALMS

CALMS, is an acronym for Collaboration, Automation, Lean, Measurement and Sharing, is helpful for analyzing DevOps structure, and its utility in an organization. They impact the modern requirements of software development lifecycle. This framework is used as a maturity model, helping managers to evaluate whether or not their organization is ready for DevOps and what changes are needed.

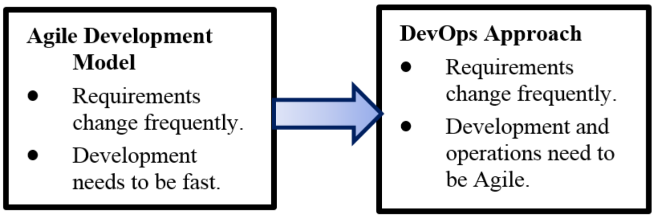
Holger Mueller, an analyst for Constellation Research, said. "The CALMS model gives a good frame of reference to compare the maturity of a DevOps team, and as such, is invaluable for assessing the state of teams for the transformational change that goes with it."



- **Culture:** It focus on customers or end users also embrace the change and experimentation. There is a culture of share responsibilities among groups and upgrade interdepartmental correspondence.
- **Automation:** Team members seek out ways to automate as many tasks as possible in order to achieve the principles of continuous delivery. For automation, infrastructure as a code is adopted for managing and provisioning computer data centers through machine readable files.
- **Lean:** Code deployments to the production environment should be small and frequent and whole applications should be developed in a way that's easy to understand. It also applies to team size; larger teams find it more difficult to agree on something.
- **Measurement:** It measures everything and show the improvement. Data is collected on everything and there are mechanisms in place that provide visibility into all systems [4]. It underpins estimation and choices in view of simple to peruse information are the way to having right options.
- **Sharing:** Sharing is essential in DevOps culture for improving the communication flow and making people work together. Therefore, it's important to share ideas, tools, findings, experiences, and thoughts within the team, among teams, and even outside the company.

7. DevOps Vs Agile:

Agile methodology, involves continuous iteration of development of the software product and testing throughout the software development lifecycle process. This software development method emphasizes on iterative, incremental, and evolutionary development. Agile development breaks the product into smaller pieces of codes and integrates them for final testing.


Agile Development Model
• Requirements change frequently.
• Development needs to be fast.
DevOps Approach
• Requirements change frequently.
• Development and operations need to be Agile.

Agile addresses gap between customer requirements and development teams whereas DevOps methodology focuses on the gap between development and operation team.

DevOps development emphasis on spreading the skill set between the development and operation teams. It also maintains consistent communication [7].

Agile puts prominence on training all the team members to have diverse of alike and equal skills. Any team member should be able to do what's required for the progress of the project, it increases understanding and bonding between them.

Agile method give priority to documentation also for future inconsistent. In the DevOps, process documentation is foremost because it will send the software to the operational team for deployment. Automation minimizes the impact of insufficient documentation. However, in the development of complex software, it's difficult to transfer all the knowledge required.

DevOps, along with automation and bug removal, contributes to creating better quality. Developers need to follow Coding and Architectural best practices to maintain quality standards whereas Agile produces better applications suites with the desired requirements. It can easily adapt according to the changes made on time, during the project life.

Agile is a set of values and principles about how to produce i.e. develop software. Example: if you have some ideas and you want to turn those ideas into working software, you can use the Agile values and principles as a way to do that. But that software might only be working on a developer's laptop or in a test environment. You want a way to quickly, easily and repeatably move that software into production infrastructure, in a safe and simple way. To do that you need DevOps tools and techniques.

III.CONCLUSION

In this review paper, we described in short, the introduction, evolution, need, and variants of DevOps. The study focuses on the knowledge map and sustainable research concerning DevOps. It demonstrates various supremacy and complexity associated with embracing DevOps, the asset subsumes more regular discharges, enhanced computerization. It likes to divide and conquer, spreading the skill set between the development and operation teams. This audit paper would apprehend the percussion in broad scale, gaining to the reckoning of DevOps.

IV. REFERENCES

[1]. J. Roche, "Adopting DevOps Practices in Quality Assurance Merging the art and science of software development," Magazine - Communications of the ACM, pp. 1–8, 2013.

[2]. SAUGATUK TECHNOLOGY, -Why DevOps Matters: practical insight on managing complex& continuous change in Microsoft, October 2014

[3]. S. Sharma and B. Coyne, DevOps For Dummies, 2015, vol. 1, no. 2. [Online]. Available: http://public.dhe.ibm.com/common/ssi/ecm/ra/en/ ram14026usen/RAM14026USEN.PDF?

[4]. M. Loukides, "What Is DevOps?" p. 15, 2012. [Online]. Available: http: //shop.oreilly.com/product/0636920026822.do

[5]. Mohamed Khaled Aljundi, -Tools and Practices to Enhance DevOps Core Values, 2016 pages 32-45.

[6]. S. Malvik, "DevOps Is Not About Trust. - DevOps.Town," 2016.

[7]. E. Minick, "DevOps - ITs Automation Revolution," 2012. [Online]. Available: http://pt.slideshare.net/Urbancode/devops-its-automation-revolution/6

[8]. S.W.Ambler. -Disciplined agile delivery and collaborative DevOps in Cutter IT Journal 24.12(2011).

Cite this article as :

Parinita Agarwal, "DevOps - A collaboration Of Developing & Deploying Software", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 4, pp. 176-181, July-August 2019. Available at doi : https://doi.org/10.32628/CSEIT195365
Journal URL : http://ijsrcseit.com/CSEIT195365