

## **DEVOPS : SPRY WAY OF IT AUTOMATION**

Opel Nahar<sup>1</sup>, Ritika Verma<sup>2</sup>, Anand Sharma<sup>3</sup>

**Abstract :** Today, innovation is accelerating and users require rapid evolution and want businesses to become agile. So as to suffice the demands of an agile business, IT operations need to deploy applications in a very repeatable, consistent and reliable manner which can only be achieved by automation. A very general impression that the word „DevOps“ provides is of unifying software development and software operations and so is it actually. DevOps is a software engineering practice whose aim is to unify software development (Dev) and software operation (Ops). Automation is a challenge and also a key aspect in IT market today. The important characteristic of the DevOps movement is to toughly support automation and monitoring at all steps of software development, from integration, testing, releasing to deployment and infrastructure management. DevOps targets at shorter development cycles, increased deployment frequency and more dependable releases, from a point of business objectives. Fundamentally, it ensures Communication, Collaboration and Integration between Developers and Operators. Using DevOps provides cultural benefits too apart from just technical benefits. DevOps is basically changing how dev and ops are done today. And it also aims to change how security is done. In this paper first we will discuss about a basic idea of DevOps, then we will describe Pre-requisites followed by advantages of DevOps and at last we will discuss the challenges.

**Keywords :** DevOps, Agile, IT tool, Automation tool.

### **1. INTRODUCTION**

DevOps isn't a product, or maybe not even a certain technology. It is actually a methodology that combines separate functions of the software development (Dev) and production and operations (Ops) into one integrated, and continuous process. It is regarding the breakdown of the bars between Dev and Ops. It influences people, processes, and technology to encourage cooperation and transformation in the entire software development and release process. Dev and Ops must not only act but also feel like they are a single team. DevOps is a cross-functional mode of working, its not a single tool, rather, it is a set of multiple tools. According to the culture (of DevOps), a single team of Engineers (developers, system admins, QA's. Testers and others turned into DevOps Engineers) have full responsibility of the Software from collecting the requirement to development, testing, infrastructure deployment, application deployment and finally monitoring & collecting feedback from the end users, making the changes again.

Apart from culturally accepting it, to implement DevOps process, different DevOps tools are required, like Puppet, Jenkins, GIT, Chef, Docker, Selenium, AWS etc in order to achieve automation at various stages which aids in accomplishing Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Monitoring to supply a software of fine quality to the customer and that too, at a fast pace. To put into effect DevOps manner apart from culturally accepting it, one also wishes diverse DevOps equipment like Puppet, Jenkins, GIT, Chef, Docker, Selenium, AWS and many others to acquire automation at different ranges with non-stop improvement, Integration, testing, Deployment and non-stop monitoring of the software to the customer at a totally speedy tempo.

### **2. PRE-REQUISITES**

You might think you are Ready for taking on or introducing DevOps into your organisation, but wait! Are You Sure You're Ready? You can start by following these four steps.

Step 1 : Align with Business Goals : Understand the wants and needs of the business side clearly, by doing this, the more better you will be able to leverage DevOps to meet those needs. Interact to the business counter-parts and figure out : How do they measure success, what are the targets for coming year, and out of them, which all goals are maximum at hazard and what exactly is holding them back.

Step 2: Characterize the present Environment :

It would be good to invest some time to characterise the current state of the organisation as thoroughly as possible. Record the configurations of every single configurable component within the organisation. Record all the information which may be needed to re-construct the starting point. One would never really do so, however, keeping all the details and things necessary to return back to the starting point is a sign that all the necessary information is present or already prepared so as to observe the changes as organisation's DevOps initiative progresses.

Step 3: Get a Buy-in From All Stakeholders :

DevOps focuses mainly on development and operations, but its implications extend throughout the organization. So, it is essential to define the processes clearly and identify the very owners who might be affected, and including them in the

<sup>1</sup> CSE Deptt., CET-MUST, Lakshmangarh

<sup>2</sup> CSE Deptt., CET-MUST, Lakshmangarh

<sup>3</sup> CSE Deptt., CET-MUST, Lakshmangarh

planning stage for sure. Set up open channels of communication. Continue to be transparent by keeping every person within the loop, as the project progresses. Doing this would increase the opportunity of success.

Step 4: Set up and Monitor Key Performance Indicators :

DevOps changes the way jobs are done in IT, because of it, to have the right metrics is crucial also as to evaluate the success. Key performance indicators (KPIs) act as a “canary in the coal mine,” giving an early warning or sign of the system when it starts to degrade.

An initial set of KPIs would possibly consist of:

Deployment frequency: Companies with DevOps cultures deploy a whole lot greater often. In fact, according to a survey from Puppet and DORA, high-performing organizations deploy 200 times more often than low performers.

Change lead time: The very definition of agility is, making changes quickly. High-performing DevOps teams average less than one hour of lead time for modifications, whilst low performers take around one to six months.

Mean time to recover (MTTR): Organizations have screw ups (failures), but DevOps companies, they re-cover very fastly, not even hours, in minutes. Precise measurements of MTTR helps the IT managers to observe the people, processes, and technology that allows fast recovery and getting rid of troubles before they can result in massive downtime.

Change fail rate: Despite of fast recovery, it is always better not to fail at all. High-performing DevOps groups have a 0–15% change failure rate. Any spike in screw ups permits IT managers to detect and solve the problem in the organization that is responsible for the rise and ensuring low change fail rates.

### 3. ADVANTAGES OF DEVOPS

**Shorter Development Cycles, Faster Innovation :** It is very difficult to evaluate if an application is ready for operations and the operations’ cycle times are extended needlessly when development and operations teams are in separate silos. With DevOps, i.e, with a combined development and operations team, applications get ready for use very quickly. This fastness of the process is very crucial in the IT market, because, the organisations succeed based on their ability to innovate faster than their competitors do. **Reduced Deployment Failures, Rollbacks, and Time to Recover :** Its mainly due to programming defects that the teams experience deployment failures. With DevOps, shorter development cycles ensures more frequent code releases, which eventually, makes figuring out code defects an easy task. Recovery time is an important issue, because some failure has to be expected but DevOps makes recovery much faster as the development and operations teams work together, exchanging ideas and accounting for both teams’ issues during development. **Improved Communication and Collaboration :** DevOps has brought a big change in software development culture. Combined teams are happier and more productive and also more focused on performance. This is happening because teams trust each other, experiment and innovate more effectively. Operations doesn’t need to wait for a different team to troubleshoot and fix a problem. The process becomes increasingly effortless as all individuals work towards a common goal. **Increased efficiencies :** Increased efficiency has led to fast development process and less prone to error. There are now ways to automate DevOps tasks. Continuous integration servers automate the process of testing code, that in turn reduces the amount of manual work needed which helps software engineers focus on completing tasks that can’t be automated. Acceleration tools also can increase efficiency. Build acceleration tools can be used to compile code more quickly. Using one environment avoids the useless task of transferring data between environments, i.e, there is no need to use one environment for development, a different environment for testing, and a third for deployment.

**Reduced Costs and IT Headcount :** DevOps benefits to reduced overall costs and IT headcount requirements. According to Kevin Murphy from Red Hat, DevOps development teams require 35 percent less IT staff and 30 percent lower IT costs.

### 4. CHALLENGES

Challenges can be Organisational challenges or Technical challenges. Challenges from Organisation point of view could be: Generally, Dev people and Ops people have clashing goals. The Operations people may get beaten into defensive mode. And by now, everyone is so used to the Dev versus Ops manner of working so it may be challenging for an Organisational point of view to adopt or bring in DevOps.

Challenges from Technical point of view could be: Operational Maturity and Technical Debt. Operational Maturity can be understood as Application versus database adjustments or changes, Microservices, Operational, testable, supportable Applications, Traceability / Visibility, Configuration, Infrastructure as code. Technical debt is basically anything about the code and the development environment that tends to slow you down. This could happen by the code being unclear or unreadable, Tangled structure & unnecessarily complicated dependencies, slow and ineffective tools, essential technical documentation that is not present or is out of date, Inessential technical documentation which is being maintained and kept updated, Long build test cycle and lack of continuous integration and there could be more depending on the kind of situation and project one is working on.

### 5. CONCLUSION

It’s Not NoOps , it is a misconception that DevOps is from the development aspect and will wipe out operations. DevOps, and its predecessors in agile operations, are being initiated out of operations teams more often than not. This is due to operation people’s existing concepts, approaches, and practices have not been in pace with what is actually required for

success. And also, it's not (just) Tools, It's not (just) Culture, It's not (just) Devs and Ops, It's not (just) A Job Title and also, it isn't everything. DevOps profits the commercial enterprises by enhancing communication, collaboration and integration of people, processes, and technologies in the IT market. Eventually, DevOps allows organizations to deliver softwares that are better, faster by improving flow, shortening, amplifying feedback loops, encouraging a culture of continuous experimentation and learning. DevOps initiatives will definitely affect the culture of the organisation. Effective communication plans, training, clear policies, and procedures, are all useful to gain the desired performance results and permit cooperation among many stakeholders that are involved in DevOps. Culture change and progress can only occur with the assistance of the people in the organisation.

## 6. REFERENCES :

- [1] <http://blogs.vmware.com/cloudops/tag/devops>
- [2] [http://en.wikipedia.org/wiki/System\\_monitoring](http://en.wikipedia.org/wiki/System_monitoring)
- [3] [http://en.wikipedia.org/wiki/Network\\_monitoring](http://en.wikipedia.org/wiki/Network_monitoring)
- [4] Chris Davis, Graphite Documentation Release 0.10.0, October 18, 2014, <http://media.readthedocs.org/pdf/graphite/latest/graphite.pdf>
- [5] [http://en.wikipedia.org/wiki/Graphite\\_%28software%29](http://en.wikipedia.org/wiki/Graphite_%28software%29)
- [6] [www.ganglia.sourceforge.net](http://www.ganglia.sourceforge.net)
- [7] [en.m.wikipedia.org/wiki/Ganglia\\_\(Software\)](http://en.m.wikipedia.org/wiki/Ganglia_(Software))
- [8] <http://sensuapp.org/>
- [9] <http://sensuapp.org/docs/0.11/handlers>
- [10] Afeez Yusuf, "Network Monitoring: Using Nagios as an example tool", May 2012.
- [11] [en.m.wikipedia.org/wiki/Cacti\\_\(Software\)](http://en.m.wikipedia.org/wiki/Cacti_(Software))
- [12] [http://www.cacti.net/downloads/docs/html/operating\\_principles.html](http://www.cacti.net/downloads/docs/html/operating_principles.html)
- [13] <http://devops.com/features/log-data-valuable-tool-devops-lifecyclebeyond/>
- [14] <https://blog.logentries.com/2014/12/connected-qa-selenium-loganalysis/>
- [15] <https://www.loggly.com/blog/loggly-qa-talking-james-urquhartcontinuous-integration-deployment-devops-role-log-monitoring/>
- [16] <http://www.splunk.com/>
- [17] <http://www.splunk.com/>
- [18] Ledion Bitincka, Archana Ganapathi, Stephen Sorkin and Steve Zhang Splunk Inc, Optimizing Data Analysis with a Semi-structured Time Series Database
- [19] <http://www.javacodegeeks.com/2015/04/log-management-tools-faceoff-splunk-vs-logstash-vs-sumo-logic.html>
- [20] <http://www.sumologic.com/>
- [21] <https://www.sumologic.com/>
- [22] <http://blog.takipi.com/the-7-log-management-tools-you-need-to-know/>
- [23] <http://cruisecontrol.sourceforge.net/overview.html>
- [24] Tim Heckel (18 Feb 2013). "Meet Travis CI: Open Source Continuous Integration". InfoQ. Retrieved 28 June 2013.
- [25] <http://java.dzone.com/articles/continuous-integration-aint>.