DevOps, Challenges and Implications

Article · October 2019

CITATIONS
0

READS
25

1 author:

Ronny Olguin
University of Murcia
1 PUBLICATION **0** CITATIONS

SEE PROFILE

DevOps, Challenges and Implications

Ronny José Olguín, Cs.B
r
onny.olguin@um.es

University of Murcia, Murcia, Spain.
October 2019

ABSTRACT

In this world of technology, organizations are requiring faster and more structured processes to accomplish the demand of the customers. It´s impossible to achieve this goal without changing the way we build, deploy and run applications. Desiring to improve their delivery process, enterprises are widely adopting DevOps.

DevOps movement is a collaborative and multidisciplinary organizational effort to automate continuous delivery of new software updates while guaranteeing their correctness and reliability [1].

Most organizations work with different standards and practices, when they try to adopt DevOps, so many challenges emerge. DevOps proposes a complementary set of agile practices to enable the
iterative delivery of software in short cycles effectively, that means they need to make a lot of changes in the development and operation teams.

Based on some resources I'm going to show you which are the most significant challenges enterprises and organizations are dealing with when adopting DevOps.

Keywords: DevOps, challenges, development, operations, agile movement, DevOps literature,

1. INTRODUCTION

DevOps and its challenges can be discussed from three perspectives: engineers, managers, and researchers. Engineers benefit from (1) qualifying themselves for a DevOps position, a technically hard task guided by over 200 papers, 230 books, and 100 tools [2]. Engineers also need to (2) learn how to re-architect their systems to embrace continuous delivery. Managers want to know (3) how to introduce DevOps into an organization and (4) how to assess the quality of already-adopted DevOps practices. Managers and engineers, also referred to as practitioners, share the necessity of choosing the best automation toolset. Finally, academic researchers (5) conduct studies to determine the state of practice in DevOps, thereby contributing to discussions among engineers and managers, and (6) educate a new generation of software engineers on DevOps principles and practices.

In this context, my research problem is devising a conceptual framework to guide engineers, managers, and academics in the exploration of DevOps tools, implications, and challenges. There are several studies and a few surveys tackling DevOps challenges. However, with few exceptions [3], they focus on a single perspective to address a given problem. In this context, this survey contributes to the field by investigating and discussing DevOps challenges from multiple perspectives: engineers, managers, and researchers. My review also explores a much broader range of sources and is more up-to-date than previous studies.

I first explain the selection and analysis procedures of these works. The selected studies are categorized and described as well. Then, I complement with non-academic sources, such as books and posts from practitioners' blogs. Also I compare my work to other reviews on DevOps.

2. BACKGROUND

Software Development was intrinsically different. Requirements were fluid to the point that they were obsolete before the features could be delivered. Business wanted more and the industry promised to deliver. [4]

The early contributors to Agile were trying to come up with ways and means of dealing with such changing requirements, but – while they made giant strides in the people and process arenas (with help from some of the stalwarts from the

manufacturing industry such as Kanban and Lean Principles) – they were severely hampered by technological limitations (or more accurately by their adopters' lack of confidence in the technologies that were available at the time).

Even so, the Software Development industry continued to inspect and adapt – and little by little it got to fixing itself. In the space from where Agile left with Lean to inform the rest of the world of a new way of thinking, Development and Operations gritted their teeth and fought through to bring the industry into its own. From this battleground, DevOps emerged as a movement that promotes the tools, processes and culture that can bring Software Development to levels of efficiency comparable to that of Manufacturing, and fulfil the promises that the industry made decades ago.

One might therefore – quite accurately – recognise it simply as the present state of the same evolutionary track that gave us Agile… but the easiest way to gain a deeper understanding of DevOps is to take in its story…

2.1 How did DevOps come about?

The story of DevOps is the story of Agile Software Development, continued.

Some time ago, there were some software developers who were unhappy with the way things were going. The engineers were mostly unhappy because they invariably had to work ridiculous hours to meet the ridiculous expectations of their leaders. The leaders were unhappy because, while the software engineers were heroes of their time and able to accomplish tremendous feats of engineering, they were – for the most part – not so able to see the tremendous commercial potential of their creations and required constant guidance and direction. The unhappy software developers tried to change things and some were more successful at it than others.

One day at the Agile 2008 conference, a discussion arose between Andrew Clay Shafer and Patrick Debois about Agile Infrastructure. Shortly after, John Allspaw and Paul Hammond delivered a typical 'thrown over the wall' routine at the 2009 Velocity conference to illustrate the Development vs. Operations notion. Later that same year, a series of events rally around the topic. The events were called 'DevOpsdays'. Debois spawned the first of these in Belgium because these were techy kind of people who called a developer (or developers, or development) 'dev', and operations 'ops' and – well – they were held on specific 'days'. During these events, the term 'DevOps' (pronounced with weird capitalisation) was made popular. It loosely translates to "things that are of varying degrees of importance to both development and operations".

Subsequent to those events – and largely unseen by mainstream business – many of the notions that have come before were integrated into something that could coherently and succinctly convey the enormous breadth of content needed to answer the original question of how software development can efficiently and effectively deliver the value that business needs – without leaders getting upset, software developers being unhappy or IT operations staff burning out. Some are more successful at this than others.

Describing the answer is still a work in progress, but even so, now – finally – DevOps is a Thing…

*"**DevOps** is a movement that was brought about to break down the walls between **Agile Development** and **IT Operations** in order to work more efficiently and be more effective".*

3. ARGUMENTS

3.1 DevOps: Implications for Engineers, Managers, and Researchers

DevOps principles, practices, and tools are changing the software industry. However, many industry practitioners, both engineers and managers, are still not aware of how their daily work can be affected by such principles, practices, and tools. By surveying the DevOps literature, I found several implications of DevOps adoption for industry practitioners and for academic researchers. My contribution is to discuss the DevOps implications for each perspective individually: engineers, managers, and researchers. In these implications, readers will find issues they will likely confront, some core concerns they should have, and potential solutions already adopted by the community. I also outline DevOps-related topics that the academic community could exploit in the future. Finally, presenting these implications raises relevant considerations, preparing my discussion of unresolved DevOps challenges. [5]

3.1.1 Implications for engineers

Here there is a list of DevOps implications affecting how engineers must architect systems, interact with their peers, and even how to adopt processes, such as incident handling:

- Microservices, cloud services, rolling back, embedded systems and IoT devices, inhibitors for high-frequency delivery, testing, quality assurance team, legacy systems, communication, learning, building the deployment pipeline, pipeline maintenance, on-boarding, incident handling, coding for stability and security…

3.1.2 Implications for managers

Based on the reviewed literature, I list implications related to how managers must face the DevOps phenomenon: required management and cultural paradigms, training people, structuring and assessing the DevOps-adoption process, as well the expected outcomes from this process:

- Adoption of lean principles, DevOps adoption, assessment, training, job titles, culture, increase in delivery throughput, building trust, building for the government…

3.1.3 Implications for researchers

Based on the reviewed literature, I list DevOps-related open topics that the academic community could exploit in future research:

- Software architecture, education, embedded systems and IoT, compliance, security, testing large-scale distributed systems, quantitative assessment metrics, deployment approach, improving interdepartmental communication, adoption strategies, investigation involving multiple organization, other research topics…

3.2 DevOps: Challenges [6]

3.2.1 Development v/s Operations

The entire concept of DevOps based on a collaborative effort from different teams to produce quick results. But, various individuals failed to understand this concept. Developers usually play with codes in order to build an innovative product. During the course of a development cycle, sometimes they do experiments to identify the best possible path. They even make changes in a quick succession. At the same time, operations team always try to achieve the highest level of quality in terms of product features and functionalities.

They always assume developers deploy untested code in production, which produce a sloppy product. This is what creates difficulties. Two teams, who supposed to work together, work on different mind-set and objectives. At the end, it creates project delays, more complexities, and lesser scope of innovation. The first challenge a company will face is to blend different people, teams together to work on a single goal.

3.2.2 Cultural Changes

Workplace culture is something which witnesses the maximum amount of changes during the phase of DevOps implementations. As DevOps is itself a culture, during the time of implementation, it collides with the existing work culture. As we told before, changes are tough. It takes a lot of time for employees to adopt a new culture and make themselves fit into that. Cultural change is not a short-term process. So, as a business owner, you need to fix this before implementing DevOps completely in your software development process. You can start with small DevOps training programmes and make your team believe in the concept. It is always a wise option to build a team of pro-DevOps who can work as influencers within the organization.

3.2.3 Tools

You can access numerous open source tools while implementing DevOps. One can feel, most of these tools are here to solve any kind of complex business challenges. During the course of development, companies allow their employees to use all these tools. But the real challenge comes when you don’t know how to use those tools. Lack of training is the main issue here. Data security will come next followed by infrastructure integration. Before addressing these challenges, you need to address your employees first. If you won’t provide tool training, be ready for more project complications, delivery delays and sloppy products. Either give training or let your team choose those tools on which they can work confidently.

3.2.4 Legacy Systems

Legacy systems are roadblocks. DevOps is a new concept and it runs on agility. Old legacy systems won't help you in getting expected DevOps results. Some of those legacy systems can't even integrate. As a result, the entire IT infrastructure will become less effective and slow. In order to overcome this challenge, you need to update with the latest technology. There are many companies who try to install new technology to work with older systems. This will make your entire IT system unproductive. You must consider legacy system and application changes at the time of ***DevOps implementation***.

3.2.5 Process Changes

If your company is following a specific guideline or framework for ***software development***, DevOps might be a tough task for you. Please remember, DevOps doesn't have any central team or fixed framework. You won't even get predefined KPIs which will help you. Challenge comes when different teams with different skill sets, work ethic and approach work together. There are high chances of conflict. Let your team choose their framework and project actions. This will reduce the chances of confusion and will empower your DevOps team.

3.2.6 Resistance to Change

We all work in our comfort zones and any change in that can make us less productive or it can also affect someone psychologically. There are people who won't be comfortable working with a different team. Collaboration is not that easy at times. As a result, this can raise a voice of distress within the organization. In past when have seen there is a misconception among employees that DevOps will take many jobs in future. But, DevOps is a methodology, which requires human efforts and implementation.
In order to come out of challenges regarding resistance to change, you can implement DevOps in a small portion of a project. Let your employees get comfortable with the change gradually. But, if you are a laggard and want to keep yourself away from such challenging situations, you can become a technically substandard company and it will affect your business process.

3.2.7 Organizational Vision

This is a common challenge various companies face. There is a central DevOps team and they work as a single team with a single supervisor. So, if any issues arise, you need to address it as a whole. But, most of the managers address issues differently. E.g. they will resolve development & operations issues separately. This will result in DevOps failure. You need to discuss issues collectively and resolve them together.

4. CONCLUSION

In this extended abstract, I have discussed DevOps challenges and implications presented in the literature I hope my readers can now better understand the impact of DevOps on daily activities based on each profile.

DevOps has come to create a culture of collaboration between the development team and the Operation (IT) team.
Among the challenges of DevOps I can mention: cultural, tools, process changes, resistance to change, organization vision challenges, and others.

Implementing DevOps in the organizations imply very extreme challenges for managers, researchers and engineers.

PERSONAL OPINION

I could say that DevOps is a set of challenges for the ones who desire to explore and work on this culture of collaboration.
Implementing DevOps in organization will help them to improve their everyday processes. Particularly I think we have to give it a try because time changes and we should do things that help not only the companies and organization but the customers. They all take advantages of DevOps.

REFERENCES

[1] Andrej Dyck, Ralf Penners, and Horst Lichter. 2015. Towards Definitions for Release Engineering and DevOps. In 2015IEEE/ACM 3rd International Workshop on Release Engineering. 3–3. Code: B26.

[2] Necco Ceresani. 2016. The Periodic Table of DevOps Tools v.2 Is Here. (2016)https://blog.xebialabs.com/ 2016/06/14/periodic-table-devops-tools-v-2/, accessed on October 2019.

[3] Steve Neely and Steve Stolt. 2013. Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy). In 2013 Agile Conference. 121–128. Code: B23.

[4] Online documentation on https://assurity.nz/archives/how-did-devops-come-about/ Last revised: October 2019

[5] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. 2019. A Survey of DevOps Concepts and Challenges. ACM Comput. Surv. 1, 1, Article 1 (January 2019), 35 pages. https://doi.org/10.1145/3359981

[6] Online documentation on https://www.apogaeis.com/blog/what-is-devops-what-are-the-challenges/ revised: October 2019