A Systematic Literature Review on DevOps Capabilities and Areas

Daniel Teixeira, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
Ruben Pereira, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
Telmo Antonio Henriques, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
Miguel Silva, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
João Faustino, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal

DOI: 10.4018/IJHCITP.2020040101

ABSTRACT

Businesses today need to respond to customer needs at an unprecedented speed. Driven by this need for speed, many companies are rushing to the DevOps movement. DevOps, the combination of Development and Operations, is a new way of thinking in the software engineering domain that recently received much attention. Since DevOps has recently been introduced as a new term and novel concept, no common understanding of what it means has yet been achieved. Therefore, the definitions of DevOps often are only partly relevant to the concept. This research presents a systematic literature review to identify the determining factors contributing to the implementation of DevOps, including the main capabilities and areas with which it evolves.

KEYWORDS

Area, Capabilities, DevOps

1. INTRODUCTION

Making any change to a business is always a complex task and usually requires investment. As such, whenever an organization adopts any new technology, methodology, or approach, that adoption must be driven by a business need. This, in turn, leads to new working methodologies for IT projects.

Traditional and modern (agile) software development usually focuses exclusively on the software development teams. In either case, once the software has been developed, it is typically handed over to the IT operations team, which takes responsibility for its deployment, ongoing maintenance and support (Jones, Noppen, & Lettice, 2016). The Agile movement has brought together programmers, testers, and business representatives. Conversely, operations teams are isolated groups that ensure stability and enhance performance by applying practices such as the Information Technology Infrastructure Library (ITIL), which equates change to risk (Hüttermann, 2012). For Debois (2011), since both development and operations serve the same customer, the needs of both must be discussed simultaneously. Treated separately, they are like separate trains
on separate tracks. No matter how fast they go, they can never meet. Due to this fact, the team commonly works in silos, which leads to a lack of information exchange. Lwakatare, Kuvaja, and Oivo (2015) say that it is impossible to effectively transmit information about all the releases performed between two different teams in continuous release mode while de França, Jeronimo, and Travassos (2016) report that Development and Operations are left to themselves and will often struggle to talk to each other, much less collaborate, and will remain mired in manual processes. The result is employees who don’t work well together, software that doesn’t work reliably, and customers that are thinking about moving to competitors (de França et al., 2016).

When two separate teams do Dev and Ops, problems can emerge and lack of synergies can occur. Separations on a technical and organizational level as well as the use of different tools have experienced an increase among Dev and Ops teams (Silva, Faustino, Pereira, & Silva, 2018). This bottleneck between Dev and Ops can affect and/or compromise the product’s quality. According to Hussaini (2014), “leveraging the critical success factors to deliver the change must have shared objectives of Dev and Ops.”. So, there is a clear disconnect as the two teams speak two different languages and have traditionally been judged according to different reward structures (McCarthy, Herger, Khan, & Belgodere, 2015).

In the midst of such evidence, DevOps emerged. DevOps applies agile and lean principles throughout the entire software supply chain. This allows a business to maximize the speed of delivery of a product or service, from the initial idea to production release and all the way up to customer feedback to improvements based on that feedback (Sharma & Coyne, 2015).

Seeing as DevOps is a recent and novel set of practices for software development (Rajkumar, Pole, Adige, & Mahanta, 2016), there is no one, agreed-upon definition of it. In a more general approach, DevOps integrates a set of characteristics and principles for software delivery that focuses on: speed of delivery, continuous testing in an environment where production takes place, being ready for shipping at any moment, continuous feedback, the ability to react to change more quickly, and teams working to accomplish a goal instead of a task (no more team boundaries causing a delay) (Sharma & Coyne, 2015). It involves an organizational paradigm shift from distributed siloed groups performing functions separately to cross-functional teams working on continuous operational feature deliveries. Instead of confining themselves to highly artificial process concepts that will never fly, organizations set up continuous delivery with small upgrades (Ebert, Gallardo, Hernantes, & Serrano, 2016).

Mohamed (2015) argues that this level of integration between development and operations will be revolutionary as releases can be driven by the business need, rather than the operational constraints. Virmani (2015) adds that this approach helps to deliver value faster and continuously, reducing problems caused by misunderstandings between team members, and help to accelerate problem resolution. In another perspective (Hüttermann, 2012), DevOps can be understood as rendering operations more agile.

Currently there is a lack of common understanding of what DevOps means for both academia and the practitioners’ communities. This knowledge gap demonstrates that there is still a need for research about the DevOps phenomenon in order to examine how it impacts software development and operations areas. Based on what has been described, there is a clear opportunity to develop a Systematic Literature Review (SLR) on the subject, with the goal of deepening our understanding of what DevOps is.

2. DEVOPS AS A DIFFERENT APPROACH

A good cooperation between IT Development and IT Operation teams are viewed to be crucial in order to ensure successful deployment and operations of IT systems (Tessem & Iden, 2008). However, for historical reasons, most IT organizations are characterized by setting clear boundaries between these two teams, which have very different goals, mindsets and cultures (Gazivoda, 2018; Swanson & Beath, 1990).
Related Content

Professional Development 1.5: Two Models for Helping Teachers Implement Digital Tools in Writing Pedagogy
www.igi-global.com/chapter/professional-development-15/137196?camid=4v1a

The Typological and Psychological Features of the Entrepreneur and Their Impact on Ethical Behavior
www.igi-global.com/chapter/the-typological-and-psychological-features-of-the-entrepreneur-and-their-impact-on-ethical-behavior/193476?camid=4v1a
Auto-Detection of Human Factor Contents on Social Media Posts Using Word2vec and Long Short-Term Memory (LSTM)
Chika Yinka-Banjo, Gafar Lekan Raji and Ifeanyi Precious Ohalete (2020). Handbook of Research on the Role of Human Factors in IT Project Management (pp. 1-13).

Success Factors and Motivators in SPI
Andreas Munk-Madsen and Peter Axel Nielsen (2011). International Journal of Human Capital and Information Technology Professionals (pp. 49-60).
www.igi-global.com/article/success-factors-motivators-spi/60527?camid=4v1a