4th International Conference on Production Economics and Project Evaluation
ICOPEV 2018, Guimarães, Portugal

UNDERSTANDING DEVOPS: EXPLORING THE ORIGINS, COMPOSITION, MERITS, AND PERILS OF A DEVOPS CAPABILITY

Catherine Crowley*, Laura McQuillan and Conor O'Brien

Innovation Value Institute (IVI), Maynooth University, Ireland

* Corresponding author: catherine.crowley@mu.ie, Maynooth University, Kilcock Road, Ireland

KEYWORDS
DevOps, Capability, Agile.

ABSTRACT

DevOps is a relatively new domain addressing the challenge of how to deploy service updates quickly and frequently, without undermining the reliability and stability of the operating environment. This paper explores the origins and composition of a DevOps IT capability. Our motivation is to understand what DevOps is, and the benefits and challenges that developing a DevOps capability presents.

DevOps encompasses key elements of processes, technology, and people. These same inter-related elements are reflected in the composition of a capability. The authors therefore consider DevOps through the lens of an enterprise capability as an enabler of DevOps deployment.

INTRODUCTION

The term DevOps was first coined in 2009 and was a response to challenges being faced by web services organizations such as Netflix, Amazon, Google, and many others. By the late 2000s, web service providers were operating large-scale complex infrastructure to support their large and growing user base. Their Operations objective was to ensure continuous user service by providing a stable and reliable operating environment. Their Development objective was to release frequent small batches of service updates and new features, using Agile development practices. The challenge, therefore, was how to deploy service updates quickly and frequently, without undermining the reliability and stability of the operating environment. The DevOps approach is to bring together Development and Operations to collaborate in solving this challenge. The scope of DevOps broadened to encompass the efficiency of the whole delivery cycle and the quality of the products and services provided.

Our motivation is to understand what DevOps is, and the benefits and challenges that developing a DevOps capability presents. Based on this motivation, we raise the following three research questions (RQs):

RQ1: How is DevOps defined or described a) in the literature and b) by practitioners?

RQ2: What are the benefits of developing a DevOps capability?

RQ3: What are the challenges when developing a DevOps capability?

An enterprise capability perspective offers a comprehensive way to consider DevOps. A capability has many elements, encompassing processes, assets and people (Peppard and Ward, 2004). DevOps affects all of these elements.

The reminder of the paper is organized as follows: First, we describe our research methodology, and then we outline the themes emerging. We explore the origins of DevOps, followed by a review of a DevOps capability. We cover the benefits and challenges of implementing a DevOps approach, and reflect on current adoption. We then examine DevOps transformation and we draw conclusions, outline the research limitations, and offer some suggestions for further research.

RESEARCH METHODOLOGY

This research is exploratory in nature. It aims to understand the emergence of DevOps through a review of current literature and interviews with practitioners from six organizations. This study adopted a dual-phase approach with respect to the research.

Phase 1- Literature Review

The authors sought to identify and analyze the key relevant themes in the stream of research and current practitioner conversation relating to DevOps. A focused literature search was undertaken to identify the key themes and their frequency. As DevOps is new, and therefore not well addressed in the academic literature through adopting a Multivocal Literature Review (MLR) approach (Garousi et al., 2016) this phase reviewed both the academic and practitioner literature to identify key themes of DevOps. Analysis of the themes was undertaken using a concept matrix approach (Webster and Watson, 2002). Synthesis of these themes with the phase 2 interview data was undertaken in the second phase. Table 1 summarizes the MLR sources.

Table 1: Multivocal Literature Review Source Types

Source	Total	Relevant to RQ's	Backward search
Academic Article	63	18	12
Report	9	3	
Blog website	27	11	
Book/ section	11	9	5
Conference	50	28	
Webinar	8	8	
Product document	12	12	
Totals	180	89	17

Phase 2- Interviews

Semi-structured interviews were conducted with six practitioners from multinational organizations. All had experience of DevOps and the capability approach, as either consultants or end users, ensuring that the interviews would produce authoritative insights from experts (Table 2). The survey instrument was a questionnaire, with 12 interview questions derived from our research questions RQ1, RQ2 and RQ3. Qualitative open-ended questions were used due to the exploratory nature of the research. This allowed for building rapport with the interviewee, clarifying and probing of the responses leading to richer data collection.

Table 2: Organizations and Interviewees Overview

Business	Staff	Role	DevOps	Code
IT Technology	100K -200K	IT Quality	2 -5	A
Management Consulting	10K-50K	IT Mgt. Consultant	2 -5	B
Banking	50K-100K	Services Manager	5 +	C
IT Technology	<10K	Enterprise Architect	1-2	D
IT Technology	100K -200K	Enterprise Architect	5 +	E
Software Consulting	<10K	Software Development	2 -5	F

To avoid bias the authors developed neutral questions and provided clear unbiased instructions during the interview (Shull et al., 2008). The interviews were recorded and transcribed and the interview transcript was sent to the interviewee to give them the opportunity to correct any errors or add any additional thoughts. The results from interviews were analyzed using qualitative coding techniques to identify the key themes as they addressed the research questions.

THEMES FROM LITERATURE AND INTERVIEWS

Themes Arising

Culture, Automation, Architecture & Modelling, Continuous Delivery and Continuous Integration, Measurement and Quality were the key themes covered in the literature. These can all be related to the acronym of C.A.L.M.S. arising in the literature, originally developed by John Willis and Damon Edwards as a means to describe DevOps, and later further refined by Jez Humble (Willis, 2010). The acronym stands for Culture, Automation, Lean, Measurement and Sharing.
The most frequently recurring themes of Culture, Automation, Architecture, Tools, Practices, Motivation, Quality and Metrics emerged in the interviews. These themes align well with those emanating from the literature. In answering the research questions, the authors integrate the interview data with the evidence from the literature in the sections below.

DEVOPS ORIGINS AND INFLUENCES

Literature

While the term ‘DevOps’ is relatively new its origins and influences have a longer history. Agile practices, technology developments in infrastructure and tools and best practices from many different approaches, such as Continuous Improvement from the 1980s, informed DevOps processes.
Agile practices increased the frequency with which new features and updates were deployed. It created a need for Operations to have robust ways to update its deployments with minimal service impacts, *“we see the customer as quite demanding"*(A), one interviewee stated. Patrick Debois held the first DevOps conference in 2009 which brought together Development and Operations IT professionals to talk about agile deployments and infrastructure (van Herpen, 2015). DevOps incorporates defined best practices including Continuous Improvement (Deming, 1982) and Theory of Constraints for prioritizing and focusing improvements (Goldratt and Cox, 1984), efficient end-to-end processes from Lean Thinking (Sharma and Coyne, 2015) and Agile development practices which introduced shorter delivery cycles delivering smaller batches of features more frequently in collaborative cross-functional teams (Highsmith and Cockburn, 2001). Concurrent Engineering from the late 1980s also advocated working simultaneously on all elements of product delivery in cross-functional teams to improve performance and quality (Poeth, 1992).
Changes in infrastructure were also an influence. In contrast to the high-cost, very large mainframes of the 1960s and earlier, technology advances bring us to the current situation where the dominant infrastructure approach is low-cost stripped-down servers, which can be mounted in banks of enclosures to provide enormous scaling capacity. With this evolution came a move away from bespoke infrastructure where automation was rarely, if ever, considered. More standardized and repeatable deployments approaches came to the fore, creating a market for automation tools. Further, in the late 1990s open source solutions were recognized as a viable basis for commercial products. The responsibility for reliability and robustness extended to become more

deeply embedded in applications. The democratization of technology and tools enabled by these advances was a key influence in the emergence of DevOps.
One of the most significant technology developments was the availability of cloud computing in the early 2000s. By enabling efficient needs-based sharing of infrastructure and computing power, cloud computing brought about the shift from a product-view to a service-view for infrastructure platforms and software solutions. Service providers now needed to deploy and update their services efficiently into the cloud. This drove the growth of DevOps through the development of enabling tools and processes, as well as establishing supportive organizational culture and structures.

Interviews

Interview data echoed many of these origins and influences. One interviewee describes the need for DevOps as "*we are all looking to be able to deliver value to the business at the speed of business"*(D).
It was recognized that many of the practices were well known and that technology developments, in particular, Cloud technology *"cloud technologies in particular, both on premises and public cloud would be very significant levers"* (B) and the open source movement, was significant in DevOps emergence. Others disagreed saying, *"There is nothing in DevOps that people haven't been doing for a long time … I think that DevOps is mostly a new age label. I think mostly driven from the early emergence of cloud" (C).*

DEVOPS CAPABILITY AND ADOPTION

Processes, Technology and People are highlighted as three central interrelated elements of DevOps (Smith-Eigenbrode, 2016; Gottesheim, 2015; Hussaini, 2015). DevOps processes concern creating a continuous efficient flow of activities from product conception through to its delivery, with embedded monitoring and measurement to inform continuous improvement (Hussaini, 2015). Technology underpins the processes in an effective chain of integrated tools to allow as much automation as possible, and a modular architecture (Chen, 2015a). People's attitudes and behaviour inform the culture; DevOps advocates a culture of collaboration, shared ownership, open communication and a learning mindset (Forsgren and Humble, 2015). The people bring their specific technical and business skills, knowledge, and experience to their role within an organizing structure which facilitates collaboration (Gottesheim, 2015).
We will look at each of these three DevOps elements in more detail in the sub-sections below and summarize key practices for each in successfully adopting DevOps.

Processes

Literature

There are a number of processes, which are central to realizing DevOps. They are encapsulated in the concept of 'Continuous Delivery', a continuous flow of features from design through to delivery. DevOps draws many processes from Agile approaches. Agile advocates Continuous Integration, Test-driven Development, and fast feedback (Mohamed, 2015). In Continuous Integration, developers merge their code into a shared versioned repository several times per day. DevOps extends this development practice into deployment and operations to establish a flow of 'Continuous Delivery'. This requires collaboration in shared processes and automation of design, code, build, test, package, deploy, release, configure and monitor (Mohamed, 2015).

Successful Continuous Delivery relies on a commitment to having a validated quality product always available for release (Humble and Farley, 2011), and reflects DevOps strong links with Lean Thinking (Fitzgerald and Stol, 2014). It requires integration and automation of as many quality assurance activities as possible in the short cycle from design to release (Ebert et al., 2016; Roche, 2013). Automation facilitates greater monitoring through metrics gathering. Automation and monitoring correlate with higher performance outcomes from DevOps (Forsgren and Humble, 2015).
Continuous Delivery is driving a move to adopt Kanban from Lean Thinking as an approach for scheduling work to help to level the daily workload (Fitzgerald and Stol, 2014). This is in contrast to the traditional staged, sequential Waterfall methodology, and is a progression from the sprint-based approach predominant in the Agile world. To gain the full benefit of the DevOps approach, organizations typically need look beyond the Development and Operations functions, adjust their processes to become product-oriented rather than project-oriented, and to reflect that in their budgeting, planning and engagement processes (Comella-Dorda et al., 2016).
The DevOps processes are encapsulated in the '3 Ways' - principles defined by Kim, Behr and Spafford (2013) when describing their DevOps journey. The first 'way' is 'System's Thinking' entailing an end-to-end system view with an emphasis on business value and quality. The second 'way' is 'Feedback Loops' with short feedback cycles and on-going improvement. (Their third way relates to a Learning Culture, which we explore later).

Interviews
The interview data largely reflects these views. *"The continuous delivery in DevOps provided a way of thinking about end to end business processes…there's an efficiency benefit because all of the hand-offs and process management effort that's typically associated with all of those separate processes has been replaced by a streamlined and automated end to end process."* (B) but *"Continuous delivery is not just a matter of automating your path to production but getting it into the governance process, in other silos, to also agree and*

contribute to the same release automated paths."(E), according to these experts. Another commented on the extent of process change required, saying that the organization had to *"turn a lot of processes inside out"* (D) to get people working together effectively to deliver business value. The point was also made that it is important to consider which processes to automate first *"we can't automate all of these processes at once, 'which are going to be the most valuables one to start with?' is very, very helpful in making sure that you get the value."(*B). Also *"agility improvements vary at different steps of the process"* and *"it's really important just to recognize the business processes as well. "* (B).
DevOps is not just a technology solution – it concerns people and process also – and processes were the things that actually drove the greatest delay.

Key DevOps Adoption Practices – Processes Perspective
The key practices to support a successful DevOps adoption that we have identified from a processes perspective include:

- ✓ *Gain commitment to have a validated quality product always available for release.*
- ✓ *Adopt a product- and service-oriented rather than project-oriented perspective.*
- ✓ *Gather metrics to inform improvement.*
- ✓ *Strong collaboration on shared processes.*
- ✓ *Look beyond development and operations for effectiveness of continuous delivery.*
- ✓ *Automation of end-to-end continuous delivery process cycle.*

Technology

Technology assets play a central role in DevOps capability. The tools used to support automation and Continuous Delivery, together with the architecture and design of the services and infrastructure are the key technology aspects to be explored.

Tools – the Literature
A Continuous Delivery flow, with short cycles, needs a high degree of automation. Automation is achieved by script writing and the use of commercial off-the-shelf tools to form an integrated chain of tools (a 'toolchain') that suits the particular organizational environment. Awareness of the role and significance of a process is essential before it is automated (Kim et al., 2013). Therefore, before selecting tools and beginning to automate, it is recommended that processes are first analyzed and refined (UpGuard, 2016; William and Murphy, 2016). Choosing the right tools to create a toolchain is an important next step (Ebert et al., 2016). The complexities of today's variety of development environments, the multiple deployment architectures that can be configured across cloud providers and in-house infrastructure provision options, and user interface devices means than there is no single tool solution. Some users have reported using 30+ tools. With appropriate architecture and design, an effective automated toolchain can be put in place to support Continuous Delivery in complex configurations (Stillwell and Coutinho, 2015).

Tools - the Interviews
The interview data revealed an emphasis on the complex, and often problematic, task of DevOps tools selection. The proliferation of tools was identified as an issue, "*I think people get very excited about the tools, they try out 47 different tools and they end up with a spaghetti*" (B), and *"I have a sense that we place an overemphasis on tools and we jump to buying a tool"* (D). The experience reported from our interviews also highlights the importance and the challenge of choosing tools and creating an automated toolchain. One stated *"So depending on the technology you use, the combination of tools varies … there is no one single tool, or even a handful of tools that constitute the toolchain" (E)* and another commented *"There are problems with tooling and toolchains where there are so many tools out there it can be difficult to know which ones are best and which ones work well with each other"* (F).

Architecture and Design – the Literature
The architecture and design of product and infrastructure impact on achieving Continuous Delivery, both in terms of automating the flow, and allowing a team to have autonomy in design and delivery of their services and features (Ebert et al., 2016; Humble and Farley, 2011). The architecture and design needs to be modular, comprising loosely coupled modules with well-defined interfaces, which are version-controlled and designed to ensure backward compatibility (Ahmed and Capretz, 2011).
A service-oriented architecture, which supports independent small services or 'microservices' meets DevOps needs for modular design (Ebert et al., 2016). It is a non-trivial task to migrate from a monolithic architecture to a modular service-oriented architecture; well-defined interfaces and adherence to backward compatibility are key design principles (Balalaie et al., 2016). Organizations can migrate their architecture over time, making adaptations step-by-step as new features are introduced (*DOES15 - Jez Humble - Architecting for Continuous Delivery*, 2015), and DevOps applications can be designed to interact with legacy systems (Schmidt, 2016). Recent infrastructure provisioning techniques like hyper-convergence support the scripted creation of contained environments. These can include database server, applications servers and webservers for test, sandbox or deployments from development containers (Abhijith et al., 2016; Kleyman, 2016). The Continuous Improvement practices and culture of DevOps can help ensure an on-going focus and effort to transform the architecture. However, it can be a challenge to get commitment for investment in

such foundational work over front-end features (Smeds et al., 2015).

Architecture and Design – the Interviews
Interviews also highlighted the significance of a modular architecture. The interdependencies between architecture and other DevOps factors was highlighted. For example, its relationship with tools and automation *"end-to-end tooling would require lots of architecture choices" (B).* The cost of establishing a supporting architecture was also acknowledged. *"So definitely the technology, architecture … is a huge investment"* (E). Architecture was highlighted as a determining factor in gaining value from DevOps *"... whether you've got a very simple architecture or you've got a complex architecture… there's a focus on value - that becomes really, really important."(B).*

Key DevOps Adoption Practices – Technology Perspective
The key practices to support a successful DevOps adoption that the authors have identified from a technology perspective include:

- ✓ *Understand the role and significance of a process before selecting tools*
- ✓ *Analyse and refine a process before automating*
- ✓ *Choose the right tools, informed by architecture and design*
- ✓ *Establish a modular architecture, with loosely coupled modules*
- ✓ *Have defined, version controlled interfaces*
- ✓ *Enact a clear and well-managed backward compatibility approach for interfaces*
- ✓ *Define a migration strategy for moving from a monolithic to modular architecture over time*
- ✓ *Gain commitment for investing in foundational architecture and design work needed*

People

People – The Literature
Underlying all aspects of DevOps are the people: the culture they create, the structures in which they operate, the skills and experience they bring, and the relationships they form are all decisive in the success of DevOps. A culture of collaboration, shared ownership and open communications, together with a commitment to experimentation and learning has been highlighted as a key enabler of DevOps (Chen, 2015a; Gottesheim, 2015; Hussaini, 2015). Research has shown that culture contributes to successful IT and organizational performance (Puppet Labs, 2014), (Puppet Labs, 2015), (Brown et al., 2016). Some organizations looking to foster a DevOps culture may need to start by building a reciprocal understanding of priorities and challenges among Development and Operations engineers (Smeds et al., 2015). For others, the focus will be on building understanding and trust throughout the organization and with customers (Fitzgerald and Stol, 2014) and extending the culture to the whole ecosystem (Comella-Dorda et al., 2016).
Organizational design and the organizing structures are significant. The creation of cross-functional (and where possible co-located) teams, a practice from Concurrent Engineering, Scrum, and Agile methodologies; is advocated in DevOps, so that a team has all the skills needed to bring a feature or service through the full lifecycle from concept to support (Erich et al., 2014). For some organizations, this requires significant re-structuring and re-location. Where co-location is not possible more effort is needed to maintain relationships and good collaboration (Smeds et al., 2015). To gain full benefit from DevOps roles and relationships need to be aligned in all organizational functions (Fitzgerald and Stol, 2014).
The sharing and development of skills and knowledge can be fostered by cross-functional teams, within a learning culture. An organization may also need to further develop their talent pool to ensure they have the expertise needed for DevOps (Smeds et al., 2015; Erich et al., 2014). Successful approaches for sharing and developing skills are to support engineer-led events and activities such as Communities of Practice and hackathons, to provide internal open source social collaboration platforms and to focus on DevOps training (Chang, 2015).

People – The Interviews
In our interviews, the cultural, organizational design and skills considerations were also a focus: "*DevOps … is a culture as much as a set of processes and tools*." (B). Interviewees reinforced the importance of cross-functional teams: "*you want to structure yourself such that you are a single team owning the entire value stream from IT inception to delivery."(E).* They highlighted that sharing skills and changing roles can be challenging and also cited political and cultural and roles tensions as hampering progress saying *"There's the cultural differences of getting actual [cross-functional] teams together, to work very closely together and that can sometimes be challenging because there can be political and cultural differences.*"(C).

Key DevOps Adoption Practices – People Perspective
The key practices to support a successful DevOps adoption that the authors have identified from a people perspective include:

- ✓ *Foster a culture of learning, collaboration, and service-orientation*
- ✓ *Build a reciprocal understanding of priorities among development and operations*
- ✓ *Build understanding and trust across the across the full product and service ecosystem*
- ✓ *Create cross-functional, co-located teams*
- ✓ *Re-structure and re-locate for collaboration*

- ✓ *Support learning and collaboration through communities of practice, hackathons, and training.*

WHAT ARE THE BENEFITS OF IMPLEMENTING DEVOPS?

The benefits of DevOps include higher organizational performance, faster and more efficient delivery, services that are more reliable and higher quality products. Customer satisfaction is increased when they receive the features they need in a timely way. The work experience of the employees is enhanced in a collaborative learning culture with levelled workloads.

Literature
The 'State of DevOps' reports (Puppet Labs, 2014), (Puppet Labs, 2015), (Brown et al., 2016), based on analysis of survey responses from more than 25,000 technical professionals worldwide, provide some compelling statistics for organizations contemplating a move to DevOps. These reports claim that companies with high-performing IT organizations are twice as likely to exceed their profitability, market share and productivity goals. They deploy 30x more frequently, with 200x shorter lead times, and they have 60x fewer failures and recover 168x faster.
Some traditional ways of software delivery operated in siloes and were sequential and slow. This resulted in development problems, long test cycles and difficult deployments. DevOps represents a way of breaking down silos, integrating development and operations to enable faster and more efficient delivery cycles to deliver higher quality product (Fitzgerald and Stol, 2014). DevOps mitigates the challenges faced by distributed software engineering and bridges gaps existing in traditional organizational processes (Mohamed, 2015). The deeper communication and mutual understanding arising from close collaboration improves cycle time and reduces costs by up to 20 percent (Ebert et al., 2016).
Dev-Ops improves product quality and ownership through a sharper focus on metrics data-driven decision-making, process standardization and Continuous Improvement initiatives (Fitzgerald and Stol, 2014; Roche, 2013). Automated test and continuous validation means problems are detected earlier, resulting in less complex problems with less time spent fixing them (Duvall, 2012). The operating environments are more stable and software is maintained in a release-ready state. DevOps provides organizations with a solution to the digital business imperative to respond and adapt to changing technologies and evolving customer needs; business value can be delivered more quickly and efficiently (Sharma and Coyne, 2015). Delivering the right product, in a timely way, with quality contributes to heightening customer satisfaction (Chen, 2015b; Forsgren and Humble, 2015; Hussaini, 2015).

Interviews
Interviewees were enthusiastic *"we hear real results, we see results"*(A), as one said. DevOps also offers more holistic benefits. The culture of collaboration, communication and learning fosters a positive workplace environment, which can contribute to engagement and motivation. One participant observed that "*understanding of the perspective and needs of others who are part of the delivery chain*" (E) caused a shift from a culture of blame to one of collaboration.
Commentators and interviewees alike indicate that the practices of DevOps offer the possibility of reducing burnout of engineers, particularly in the operational domain where "*a focus on things like automated continuous testing and integration would free up a huge amount of valuable time*" (D) thereby eliminating some of the time pressures contributing to burnout and offering "*a less stressful way of working*." (D).
One interviewee noted, "*The biggest benefits opportunity is when you combine together four different topics, so agile software development, continuous delivery, cloud technologies and simplification of the landscape."(B)*.
Another commented, "*I really think that it's helped more companies release more regularly… it's about getting it out there with confidence."* (F). They also highlighted that benefits only accrue if DevOps suits the environment: *"It's about how stable I think an IT solution is. So how much change do you expect, how volatile is the environment? If it's stable and not a lot of enhancements are expected, I don't see a need for the DevOps approach."(A).*

GUIDING A DEVOPS TRANSFORMATION

For an organization to reap the many benefits of DevOps, it needs to evaluate and address the challenges that DevOps adoption entails. The first step is to gain a clear knowledge of its core principles and understand what benefits it can bring to an organization (Smith-Eigenbrode, 2016). Fundamental to success is understanding that DevOps is not a set of tools, nor an automation task, nor creating teams made up of both developers and operations engineers; it is a combination of all of that and more; each organization will have its unique context and must identify the benefits it values and develop its own approach to achieving DevOps (Ebert et al., 2016; Smith-Eigenbrode, 2016; McCarthy et al., 2015; Smeds et al., 2015). As stated by one interviewee "*The key thing is - it's all about business strategy, it's all about understanding what's important to your business and making sure that you know what's going to be of value to you* "*(F).*
There are a several factors that facilitate the success of any transformation program: a clear vision, strong and persistent executive sponsorship, adequate resources, open and extensive communication, empowered participation of those who will own or are impacted by the change and building understanding and commitment among all stakeholders (Armenakis and Harris, 2009). Significant change takes time, so planning short- and

long-term goals, on-going evaluation of progress and sustained commitment throughout the program all aid success (Sirkin et al., 2014; Kotter, 2007). Many of these factors are echoed in reports on DevOps adoption experiences, for example, the importance of executive support (Schmidt, 2016; Brown et al., 2016; Erich et al., 2014; Chang, 2015; Schmidt, 2016; Kotter, 2007; Sirkin et al., 2014, Balalaie et al., 2016) and the need for a plan to empower those who own the change (Chang, 2015). DevOps has principles that support on-going change: a focus on metrics and measurement to guide improvement and the emphasis on Continuous Improvement both foster change that will in turn strengthen DevOps capability, and contribute to creating a learning culture. Metrics can also help to communicate success, and can be used to build support. For broad organizational support, it is recommended to have metrics that have meaning in terms of business value, and to measure key activities and investments. Recommended categories of business metrics include: productivity, for example shorter time to market; quality, one possible metric being increased reliability of solutions; operating expense such as a measure of cost avoidance; and capital expense such as a metric on the improved utilization of infrastructure (Elliot, 2014). With appropriate metrics and understanding of the feedback, it is possible to fail fast, learn and recover quickly (Smith-Eigenbrode, 2016) and shorten the path to business value.

CURRENT DEVOPS ADOPTION AND DEVELOPMENTS

Cloud and Web development organizations were early adopters of DevOps and they have acted as guides for others (Ebert et al., 2016; Hintsch et al., 2015) and drove their competitors to adopt similar approaches. They contributed to the impetus for broader adoption of DevOps across the technology sector (e.g. in finance and telecommunications), and the range of infrastructure and tools available provide a technical foundation to support broad adoption. When asked about their motivation to adopt DevOps one interviewee stated simply *for me it is a matter of survival."(E)* Moreover, another cited "*the prospect of a less stressful way of working*" (D) in addition to the expected "*reduce delivery cycle time*."(E).

The interest in DevOps has never been higher, although the extent of DevOps adoption is varied. Some organizations are in exploration and piloting phases. For others DevOps informs all activities, and they are leveraging the full power of the available technology (cloud, microservices design, fully automated processes) to develop and operate extremely complex, scalable and highly resilient systems.

There is growing acknowledgement that integration needs to extend beyond Development and Operations to all business functions, such as Finance and HR. In Finance for example, financial governance would need to be decoupled from funding cycles so that DevOps teams can continue working as long as they are achieving organizational results (Rudder, 2015). The term 'BizDev' has been used to represent this integration (Fitzgerald and Stol, 2014). It is expected that in the near future DevOps teams will bring security, compliance, and audit teams into the project-planning cycle to embed these requirements in their automated processes in order to reduce security risks and other business risks (Elliot, 2014); a view also highlighted in our interviews. Studies have shown that Agile practices can be successfully used in regulated industries. Some envision a future where team members across the organization will transcend traditional roles and organizational boundaries, and stakeholders will have access to real-time information from both business and IT systems to better guide the organization (Rudder, 2015). However, organizations need to carefully consider the meaning of DevOps for them, understand what investment and effort adoption would require, and most importantly be clear on the value and benefit it offers in their particular context.

CONCLUSIONS

DevOps is a relatively new phenomenon. Our research documents the origins and composition of a DevOps IT capability. In response to RQ1, we have identified typical DevOps drivers and outcomes, and key DevOps adoption practices from process, technology and people perspectives. We have answered RQ2 and RQ3 by identifying the benefits to be expected from adoption and the challenges to be faced. The research suggests that the adoption of DevOps has been a positive move for software development. In the interviews no one suggested that they would advise against DevOps with one suggesting DevOps adoption is the less risky strategy, stating *"I think that's actually the bigger risk is not accepting the fact that it [DevOps] is there and then working out how to manage it; because if you have software, you have DevOps*."(C). All claimed benefit to varying degrees. The literature also supports this. However, it is unclear where the value is coming from – can this be attributed to DevOps alone or are there other forces also contributing? Was the reported improvement inevitable? We believe that there has not yet been rigorous testing of the claims made by practitioners. Empirical evidence needs to be gathered to support the view that the positive benefits observed emanate from DevOps alone. It is also clear that to gain maximum benefits, the DevOps approach needs to be expanded to the rest of the organization. Otherwise, the organization is limited by what the other functions or departments can support.

Limitations

As DevOps is new and not well addressed in the academic literature, the authors adopted a MLR to gain a fuller understanding of the topic. We acknowledge that this is a novel approach. We view this practical approach both as a limitation and as an innovation in expanding the source horizons when exploring new and emerging topics. Another limitation of this research is the low number of interviewees. However, the interviewees experience base with DevOps spanned multiple companies and they provided rich data on their experiences of DevOps in practice.

Recommendations for further research

Although the literature on DevOps is very positive, and interviewees also express satisfaction, it would be valuable to have stronger evidence to support these assertions. We identified a need for empirical studies to validate the benefits of DevOps in practice. Some questions which future research on DevOps could focus on include:

- Are the tools delivering the efficiency and effectiveness expected?
- What are the implications for product quality and regulatory compliance management?
- How far can DevOps adoption progress within a waterfall approach?

REFERENCES

Abhijith, U., Taranisen, M., Kumar, A. and Muddi, L. (2016) ‘The efficient use of Storage Resources in SAN for Storage Tiering and Caching’, IEEE, pp. 118–122.
Ahmed, F. and Capretz, L. (2011) ‘An architecture process maturity model of software product line engineering’, *Innovations in Systems and Software Engineering*, vol. 7, no. 3, pp. 191–207
DOES15 - Jez Humble - Architecting for Continuous Delivery (2015) [Film].
Armenakis, A. A. and Harris, S. G. (2009) ‘Reflections: our Journey in Organizational Change Research and Practice’, *Journal of Change Management*, vol. 9, no. 2, pp. 127–142.
Balalaie, A., Heydarnoori, A. and Jamshidi, P. (2016) ‘Microservices Architecture Enables DevOps: an Experience Report on Migration to a Cloud-Native Architecture’, *IEEE Software*.
Brookes, N. and Backhouse, C. (1997) ‘Implementing concurrent engineering in different environments: factors for success’, IET, pp. 153–159.
Brown, A., Forsgren, N., Humble, J., Kersten, N. and Kim, G. (2016) *2016 State of DevOps Report*, Puppet labs.
Chang, S. (2015) ‘Intel’s Journey to Large Scale DevOps Transformation’,.
Chen, L. (2015a) ‘Continuous Delivery: Huge Benefits, but Challenges Too’, *IEEE Software*, vol. 32, no. 2, pp. 50–54
Chen, L. (2015b) ‘Towards Architecting for Continuous Delivery’, Bass L., Lago P., and Kruchten P. (eds), *Proceedings - 12th Working IEEE/IFIP Conference on Software Architecture, WICSA 2015*, Institute of Electrical and Electronics Engineers Inc., pp. 131–134
Comella-Dorda, S., Lohiya, S. and Speksnijder, G. (2016) *An operating model for company-wide agile development | McKinsey & Company*
Deming, W. E. (1982) *Out of the Crisis*, Cambridge, MA; London, England, The MIT Press.
Duvall, P. (2012) ‘Breaking down barriers and reducing cycle times with devops and continuous delivery’, *Gigaom Pro*.
Ebert, C., Gallardo, G., Hernantes, J. and Serrano, N. (2016) ‘DevOps’, *IEEE software*, vol. 33, no. 3, pp. 94–100
Elliot, S. (2014) *DevOps and the cost of downtime: Fortune 1000 Best Practice Metrics Quantified*, Insights
Erich, F., Amrit, C. and Daneva, M. (2014) ‘Report: Devops literature review’, *University of Twente, Tech. Rep*.
Fitzgerald, B. and Stol, K.-J. (2014) ‘Continuous Software Engineering and Beyond: Trends and Challenges’, *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, NY, U.S.A., ACM
Forsgren, N. and Humble, J. (2015) *DevOps: Profiles in ITSM Performance and Contributing Factors*, SSRN Scholarly Paper, Rochester, NY, Social Science Research Network
Garousi, V., Felderer, M. and Mäntylä, M. V. (2016) ‘The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature’, ACM Press, pp. 1–6
Goldratt, E. M. and Cox, J. (1984) *The goal: excellence in manufacturing*, North River Press.
Gottesheim, W. (2015) ‘Challenges, benefits and best practices of performance focused DevOps’, *LT 2015 - Proceedings of the 4th ACM/SPEC International Workshop on Large-Scale Testing, in Conjunction with ICPE 2015*, Association for Computing Machinery, Inc, p. 3
van Herpen, D. (2015) ‘DevOps - in 3 minutes’, *Blog | Van Haren Publishing* [Online].
Highsmith, J. and Cockburn, A. (2001) ‘Agile Software Development: The Business of Innovation’, *Computer*, vol. 2001, no. September.
Hintsch, J., Schrödl, H., Scheruhn, H.-J. and Turowski, K. (2015) ‘Industrialization in Cloud Computing with Enterprise Systems: Order-to-Cash Automation for SaaS Products’, [Online].
Humble, J. and Farley, D. (2011) *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Kindle Edition. Boston, MA 02116, U.S.A., Pearson Education, Inc.
Hussaini, S. W. (2015) ‘A Systemic Approach to Reinforce Development and Operations Functions in Delivering an Organizational Program’, *Procedia Computer Science*, vol. 61, pp. 261–266.
Kim, G., Behr, K. and Spafford, G. (2013) *The Phoenix Project*, Kindle. Portland, Oregon, IT Revolution Press.
Kleyman, B. (2016) ‘What Cisco’s New Hyperconverged Infrastructure Is and Isn’t Good At’, Data Centre Knowledge [Online].
Kotter, J. P. (2007) ‘Leading Change Why Transformation Efforts Fail’, *Harvard Business Review*, vol. 2007, no. January, pp. 96–103.
McCarthy, M. A., Herger, L. M., Khan, S. M. and Belgodere, B. M. (2015) ‘Composable DevOps: Automated Ontology Based DevOps Maturity Analysis’, Chou W., P. I., Maglio P. P. (ed), *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, Institute of Electrical and Electronics Engineers Inc., pp. 600–607.
Mohamed, S. I. (2015) ‘DevOps shifting software engineering strategy value based perspective’, *Journal of Computer Engineering (IOSR-JCE)*, vol. 17, no. 2, pp. 51–57 [Online].
Peppard, J. and Ward, J. (2004) ‘Beyond strategic information systems: towards an IS capability’, *The Journal of Strategic Information Systems*, vol. 13, no. 2, pp. 167–194 [Online].
Poeth, D. F. (1992) ‘Concurrent engineering-key to cost-effective product reliability, maintainability, and manufacturability’, IEEE, pp. 131–134.
Puppet Labs (2014) *2014 State of DevOps Report*, Puppet Labs & IT Revolution [Online]. (Accessed 12 May 2016).
Puppet Labs (2015) *2015 State of DevOps Report*, Puppet

Labs & IT Revolution (Sponsored by PWC) [Online]. (Accessed 2 June 2016).
Roche, J. (2013) 'Adopting DevOps Practices in Quality Assurance', *Commun. ACM*, vol. 56, no. 11, pp. 38–43
Rudder, C. (2015) *Creating a culture for DevOps to thrive | The Enterprisers Project* [Online].
Schmidt, M. (2016) '5 Prerequisites for a Successful DevOps Initiative', *DevOps.com* [Online].
Sharma, S. and Coyne, B. (2015) *DevOps for Dummies*, Hoboken, NJ 07030, U.S.A., John Wiley and Sons Inc.
Shull, F., Singer, J. and Sjøberg, D. I. (2008) *Guide to advanced empirical software engineering*, Springer, vol. 93.
Sirkin, H. L., Keenan, P. and Jackson, A. (2014) 'The hard side of change management', *IEEE Engineering Management Review*, vol. 42, no. 4, pp. 132–132 [Online].
Smeds, J., Nybom, K. and Porres, I. (2015) 'DevOps: A Definition and Perceived Adoption Impediments', in *Agile Processes in Software Engineering and Extreme Programming*, Springer, pp. 166–177.
Smith-Eigenbrode (2016) *5 Things DevOps is Not - DevOps.com* [Online].
Stillwell, M. and Coutinho, J. G. (2015) 'A DevOps approach to integration of software components in an EU research project', ACM, pp. 1–6.
UpGuard (2016) *DevOps for Cynics*, UpGuard [Online].
Webster, J. and Watson, R. T. (2002) 'Analyzing the Past to Prepare for the Future: Writing a Literature Review', *MIS Quarterly*, vol. 26, no. 2, pp. xiii–xxiii.
William, D. P. and Murphy, T. E. (2016) 'Avoid Failure by Developing a Toolchain That Enables DevOps', Gartner [Online].
Willis, J. (2010) *What Devops Means to Me* [Online].