How DevOps capabilities leverage firm competitive advantage: A systematic review of empirical evidence

Olivia H. Plant
Industrial Engineering and Business Information Systems
University of Twente
Enschede, The Netherlands
and
Quint Nederland B.V.
Amstelveen, The Netherlands
o.h.plant@utwente.nl

Jos van Hillegersberg
Industrial Engineering and Business Information Systems
University of Twente
Enschede, The Netherlands
j.vanhillegersberg@utwente.nl

Adina Aldea
Industrial Engineering and Business Information Systems
University of Twente
Enschede, The Netherlands
a.i.aldea@utwente.nl

2021 IEEE 23rd Conference on Business Informatics (CBI) | 978-1-6654-2069-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/CBI52690.2021.00025

Abstract—**DevOps is an agile software delivery approach which combines IT development and operation functions into cross-functional teams and promotes team autonomy and automation of processes. Many companies transform their IT departments according to the DevOps paradigm, hoping to increase their pace of software delivery and achieve a tighter collaboration between business and IT functions. Whereas reasons for adopting DevOps are often described in terms of operational efficiency, the paper at hand aims to investigate whether implementing DevOps can additionally contribute to the strategic advantage of companies. To this end, we have conducted a systematic review of 37 empirical research papers on DevOps capabilities and analyzed the results in the context of the dynamic capabilities theory. Our conceptual model proposes that DevOps teams can contribute to firm competitive advantage by building both business- and technology-related capabilities which enable them to sense market opportunities, make fast and targeted decisions and transform their assets in case of changing circumstances. This research aims to generate a deeper understanding of the impact which software delivery approaches like DevOps can have on firm competitive advantage and provides insight into the sources of dynamic capabilities in modern IT organizations.**

Keywords—DevOps, dynamic capabilities, IT capabilities, firm competitive advantage

I. INTRODUCTION

Agile software development methods are gaining popularity in the industry. Their main advantage, as opposed to traditional development approaches, is their ability to respond to rapidly changing environments [1] which makes them an attractive operating model for many companies. Additionally, product-oriented, agile teams are deemed to handle customer demands better [2], although an increasing disconnect between IT development and operation functions presents a bottleneck in this process and impedes the fast delivery of new software functionality [1]. Organizations therefore turn to software delivery approaches like DevOps which aims to bridge this gap by combining both IT development and operations into cross-functional teams [2], [3]. Agile and DevOps are considered important methods for companies undergoing a digital transformation since they enable the delivery of innovative products or services via digital services platforms [4].

Although other literature reviews on DevOps capabilities have been performed prior to our research [5]–[7], the wider implications of using DevOps in organizations are not yet fully understood and the link between DevOps and firm competitive advantage has been scarcely researched until now. To the best of our knowledge, no theory is available that explains how DevOps teams can deliver strategic value to their organizations. In research and practice, the benefits of DevOps are often described and measured in terms of operational efficiency instead of strategic impact, focusing on metrics like code deployment frequency and lead time for changes [8]. The research at hand intends to further build on the available literature by explaining how exactly DevOps teams can contribute to organizational advantage and by presenting a conceptual model for this. This premise is supported by Wiedemann and Schulz [9] who have adopted a resource-based view and concluded that DevOps teams can build seven key capabilities which enable them to perform software process innovation which in turn contributes to competitive advantage.

Extant literature has demonstrated that organizational agility leads to superior firm performance and is strongly related to the presence of IT capabilities [10]. This relationship is mediated by the fact that IT competences allow companies to build digital platforms which in turn support organizational agility [10]. In order to enable these IT capabilities, we argue that it is important for researchers to investigate contemporary software delivery approaches like DevOps in more detail. This is expected to enhance the current understanding on how DevOps contributes to agility on a technology- and organizational level and thus may leverage competitive advantage. Furthermore, the research at hand intends to provide a deeper understanding how the digital transformation affects the sources of competitive advantage.

We build on the theory of dynamic capabilities [11], [12] which is an enabler to organizational agility [13], [14]. The dynamic capabilities framework was selected as the basis of our research due to its significant contribution and frequent application to information systems (IS) and strategic research. To this end, we conduct a systematic literature review [15] of empirical research on DevOps implementations and adoptions to identify relevant DevOps

This research was supported by Quint Nederland B.V., De Oude Molen 1, 1184 VW Amstelveen, The Netherlands.

978-1-6654-2069-3/21/$31.00 ©2021 IEEE
DOI 10.1109/CBI52690.2021.00025

capabilities. We then analyze these in the context of the dynamic capabilities theory [12].

The research at hand therefore aims to answer the following research question:

How does the DevOps approach enable software delivery teams to build dynamic IT capabilities which contribute to firm competitive advantage?

The remainder of this paper is structured as follows: Section II provides more information on the DevOps approach and on the dynamic capabilities theory. Section III describes our literature review approach in detail. Section IV summarizes the results and presents our conceptual model. Section V discusses the results and section VI concludes the paper.

II. Background

A. *DevOps*

DevOps is used as an umbrella term to describe approaches in which organizations combine IT operations and development responsibilities into cross-functional teams [16]. Reasons for adopting DevOps range from wanting to improve the pace and quality of software deployment to improving collaboration between development and operations functions [16]. Furthermore, product-oriented, agile teams are deemed to be more effective at handling rapidly changing customer demands [2].

DevOps is an ambiguous concept. It has been referred to as many different things such as a movement, a philosophy, a practice, a culture or a mindset. Furthermore, there are tensions as to whether DevOps is mainly about the cultural aspects or is more of a technical solution [17]. Literature reviews have shown that there is no uniform definition of DevOps [16], [18] although various studies have defined some general patterns that DevOps processes usually share. Smeds, Nybom and Porres defined DevOps as *"a set of engineering process capabilities supported by certain cultural and technological enablers"* [19]. We select this definition as a guiding principle in our research due to its specific focus on process capabilities. Furthermore, the aforementioned characterization is broad enough to account for varying interpretations of the DevOps phenomenon in literature. Although DevOps may thus be interpreted in slightly different ways in extant literature, we argue that the unique combination of development and operation activities yet calls for a similar set of capabilities which these teams need to develop.

A frequently used summary of DevOps principles is the acronym CAMS which represents the dimension of culture, automation, measurement and sharing [3], [20]. These will be summarized shortly in the following: First, DevOps involves decentralized decision-making structures and DevOps teams therefore need great autonomy [21] in order to react adequately and timely to new demands and incidents. Moving towards DevOps therefore requires companies to not only change organizational structures and invest in new technology but primarily calls for the adoption of a new organizational culture. Furthermore DevOps teams aim to automate most of their processes by using principles such as continuous delivery in which software code changes are automatically tested and staged [20]. The measurement concept describes the effort to continuously monitor business metrics and key performance indicators [20]. Finally, DevOps teams share knowledge, tools and infrastructure with each other [3], [20].

B. *Dynamic Capabilities*

The question how some companies yield competitive advantage above others has long been discussed in strategic management literature. The concept of dynamic capabilities [11], [12], [22] was introduced as an extension to the resource-based view which aims to explain competitive advantage in terms of the resources which an organization accumulates. This paradigm is opposed to strategic models which emphasize the exploitation of market power like the competitive forces model and the strategic conflict model [11].

Recently, Brosig, Westner, and Strahringer [23] concluded that the resource-based view is the most dominantly used perspective in IT capabilities research and that contemporary research additionally incorporates technology-based views in their IT capability conceptualizations. However, the resource-based view alone has not sufficiently explained the advantage of firms in rapidly changing environments [22]. According to Teece, Pisano, and Shuen [11], accumulating resources alone is not sufficient, in order to sustain significant competitive advantage. They further argue that winners in the global marketplace have demonstrated responsiveness and rapid product innovation and were able to effectively coordinate and redeploy internal and external competences. They therefore define dynamic capabilities to be *"the firm's ability to integrate, build and reconfigure internal and external competences to address rapidly changing environments"* [11].

Although more recent business and enterprise frameworks have also adopted capability-based transformation approaches, we select the dynamic capabilities theory [11] as a fundament for our research due to its significance in the strategy and IS research domains and its potential to explain the link between capabilities and firm competitive advantage in detail.

Dynamic capabilities reside in the organizations managerial processes and structures [11]. These are in turn determined and shaped by the firm's assets and the unique paths which are available to the organizations. These paths are largely determined by the history of the organization and its technological opportunities. The authors therefore argue that dynamic capabilities are unique and idiosyncratic to an organization and need to be difficult to replicate. Superior operational efficiency is valuable but not a dynamic capability [12]. Contrary to this, Eisenhardt and Martin [22] argue that dynamic capabilities are in fact idiosyncratic in their details but at a higher level constitute a set of specific and identifiable processes. Examples of such processes are product development, strategic decision making and alliancing [22].

Teece [12] has categorized dynamic capabilities into three classes. Each of these capabilities is undergirded by organizational and managerial processes, structures and systems. The first type of dynamic capabilities enables an organization to *sense* and shape new opportunities and threats. These can either be of technological nature or relate to the market in which the organization is operating. Secondly, organizations require capabilities to *seize* the identified opportunities and address it with new products or services. This requires managers to make swift and targeted investment decisions. The last set of capabilities enables the firm to *manage threats and transform* the organization. This involves recombining and configuring the asset base and organizational structures as a consequence of changing markets and organizational growth.

As previously stated, the resource-based view and more specifically dynamic capabilities have been frequently used in IS research. IT assets and capabilities are generally seen as valuable enablers and contributors to enterprise dynamic capabilities. Wade and Hulland [24] argue that *"IS resources may take on many of the attributes of dynamic resources, and thus may be particularly useful to firms operating in rapidly changing environments."* They furthermore believe that IS resources might not directly lead to competitive advantage but yet remain critical to ensuring organizational competitiveness in volatile environments.

III. Review Protocol

In order to answer our research question, we adopted a systematic literature review approach [15]. Figure 1 delineates the steps pertaining to this process as suggested by Rouhani, Mahrin, Nikpay, Ahmad, and Nikfard [25].

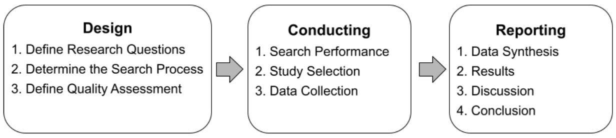

Figure 1. Systematic literature review stages as suggested by Rouhani, Mahrin, Nikpay, Ahmad, and Nikfard [25]

We have introduced our research rationale and research questions in the introduction of this paper. Following this, we started with an exploratory literature search which served as a basis for our review protocol which is an important element of any systematic literature review [26].

This protocol contained the keywords for the literature search, the databases and search strategy, as well as inclusion and exclusion criteria for selecting suitable literature. All of these criteria will be discussed in the following sections in which we will describe each step of the conducting phase separately. It is of paramount importance that this process is documented rigorously, in order to show credibility and allow other researchers to assess the exhaustiveness of the research [27].

A. *Search performance*

We initially conducted an exploratory literature search to identify suitable databases, keywords and selection criteria that would provide us with relevant search results. The initial search terms were then refined by scanning relevant papers and identifying often used keywords. Priory tested search terms such as "DevOps" proved to be too broad and were therefore narrowed down. This process led to the search terms as shown in TABLE I.

TABLE I. Keywords for database search

	AND	
DevOps	Capabilit*	***OR***
	Enabler*	
	Concept*	
	Asset*	
	Resource*	
	Dimension*	
	Practice*	

The final search was conducted at the end of December 2020. These keywords were entered into six different databases which are documented in TABLE II. These databases were also identified based on the exploratory review and were part of our review protocol. They are widely recognized sources in the field of information systems research. Whereas the search parameters were the same throughout all databases, the syntax of the search had to be adjusted slightly according to the query language of the database. The search parameters were narrowed to the title, abstract and keywords of the articles since a full-text search yielded too many irrelevant results.

TABLE II. Initial search results

Search engine	**Sub-library (if applicable)**	**Results**
Scopus		692
Web of Science	WoS core collection	366
IEEE Xplore		285
ACM Digital Library	Full text collection	137
Science Direct		41
AISeL	AIS electronic library	22
Total		**1543**

B. *Study selection*

The study selection and data extraction steps were performed by the first author. The search process is demonstrated in Figure 2 according to the guidelines of the PRISMA statement for systematic reviews [28].

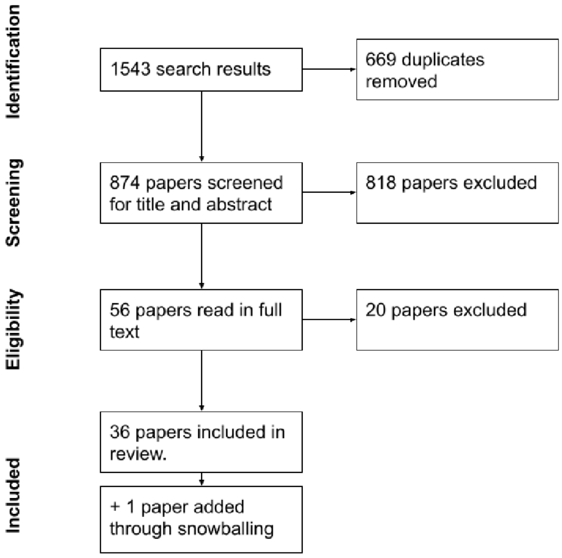
Identification
Screening
Eligibility
Included
1543 search results
669 duplicates removed
874 papers screened for title and abstract
818 papers excluded
56 papers read in full text
20 papers excluded
36 papers included in review.
+ 1 paper added through snowballing

Figure 2. Information flow diagram according to PRISMA [28].

The database search amounted to 1543 raw results. These were firstly filtered for duplicates. These results where then scanned based on their title and abstract to determine whether they met the inclusion or exclusion criteria as defined in the review protocol. These criteria are shown in TABLE III.

TABLE III. INCLUSION AND EXCLUSION CRITERIA.

Inclusion	Investigates practices, concepts, capabilities or assets related to (Biz/Sec)DevOps
	The paper contains primary research, thus it contains empirical data either as input or validation of the theory
Exclusion	Paper only focusses on DevOps tooling or architecture, e.g. infrastructure provisioning, CI/CD, or microservices
	Paper focusses on educating students about DevOps
	Paper focusses on implementation of DevOps practices in a very narrow context (e.g. embedded systems, IoT)
	Literature reviews

Papers which obviously did not contribute to the research question were eliminated. Furthermore, only papers that contained an empirical research component were taken into account which means that systematic literature reviews and expert opinions based on theory were not selected. This decision was made to ensure that the capabilities which were to be extracted from the papers were all observed in practice and were not based on implicit expectations and assumptions.

Another important criterion was the fact that the papers needed to address the DevOps concept from an organizational perspective instead of viewing it as a purely technical approach. The rationale for this decision was that the literature stream which focused on DevOps toolchains and pipelines did not research DevOps capabilities in relation to their organizational context. It was therefore not deemed feasible to identify the strategic value of these capabilities against this background.

This selection process resulted in 56 eligible papers which were read in full text. We then excluded another 20 papers since they did not meet the criteria. One paper was added by evaluating forward and backwards references of the included papers [29]. This process resulted in a final sample of 37 papers.

As discussed above, our main quality condition was the inclusion criterion that all papers needed to contain empirical evidence which means that they contain evidence from e.g. case studies, interviews or practitioner experiences as input or validation of the research. This is in line with the study design hierarchy by Kitchenham [26] who notes that expert opinions based on theory or consensus constitute a low-quality study design. Two papers in our sample were based on systematic literature reviews but were validated by industry experts based on their experiences. Due to the fact that DevOps is still a relatively young research field with a limited number of publications, we have decided to not filter the papers further based on their publication type or outlet.

As shown in Figure 3, all included paper were published between 2014 and 2020. The higher number of included publications from the past few years is in line with the overall increasing trend of research on DevOps. 26 of the papers were conference papers and 11 papers were journal publications. The fact that most of the journal papers stem from recent years furthermore shows that the maturity of research on DevOps is slowly increasing.

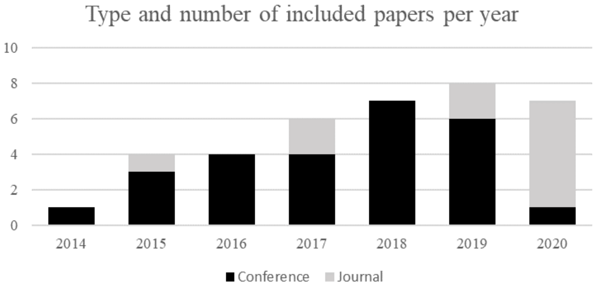

Figure 3. Type and number of included papers per year.

C. *Data collection & synthesis*

The data was extracted and synthesized by drawing on elements from two recognized yet contrasting qualitative evidence synthesis methods. Similar to the study selection procedure, these steps were performed by the first author. We firstly applied a grounded theory approach which allows researchers to establish new synthetic constructs that exceed the level of individual studies [30]. To this end, the empirical evidence given in all 37 papers was coded by using the qualitative data analysis tool AtlasTI. We applied labels to chunks of text which mentioned or implied capabilities and practices that were relevant in a DevOps context. This process amounted in 126 separate codes. During this process, we applied comparative analysis, which means that the codes were continuously compared, merged or renamed and relationships between codes as described in the literature were defined [31]. Simultaneously, we grouped the single codes into broader categories of codes. The final overview resulted in a list of specific DevOps practices (codes) and a more comprehensive list of DevOps capabilities (code categories).

During the second step of our data synthesis, we mapped the code categories to the dynamic capabilities framework by Teece [12]. In doing so, we used features of the framework synthesis approach in which an a priori framework is selected which provides the thematic categories to which data can be coded [32]. As opposed to grounded theory, which is inductive in nature, framework synthesis methods are realist, deductive and build on existing models and frameworks [33]. Framework synthesis is therefore very suitable to test the extent to which an existing theory is supported by empirical evidence [30], which is in line with our research question.

IV. RESULTS

In addition to the three classes of dynamic capabilities as defined by Teece [12], we identified two additional sets of capabilities that are relevant for DevOps teams to achieving competitive advantage. These sets serve as enablers and outcomes of the dynamic DevOps team capabilities. We name the first set of capabilities *Organizational enabler capabilities* since they are primarily governed on an organizational level. These capabilities support the DevOps teams in their way of working and should therefore already be in place before building other capabilities.

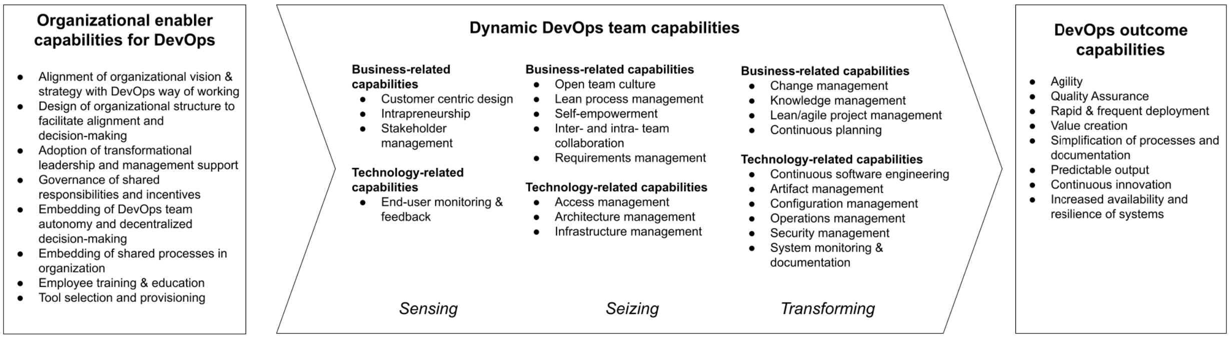
Organizational enabler capabilities for DevOps
Alignment of organizational vision & strategy with DevOps way of working
Design of organizational structure to facilitate alignment and decision-making
Adoption of transformational leadership and management support
Governance of shared responsibilities and incentives
Embedding of DevOps team autonomy and decentralized decision-making
Embedding of shared processes in organization
Employee training & education
Tool selection and provisioning
Dynamic DevOps team capabilities
Business-related capabilities
Customer centric design
Intrapreneurship
Stakeholder management
Technology-related capabilities
End-user monitoring & feedback
Business-related capabilities
Open team culture
Lean process management
Self-empowerment
Inter- and intra- team collaboration
Requirements management
Technology-related capabilities
Access management
Architecture management
Infrastructure management
Business-related capabilities
Change management
Knowledge management
Lean/agile project management
Continuous planning
Technology-related capabilities
Continuous software engineering
Artifact management
Configuration management
Operations management
Security management
System monitoring & documentation
Sensing
Seizing
Transforming
DevOps outcome capabilities
Agility
Quality Assurance
Rapid & frequent deployment
Value creation
Simplification of processes and documentation
Predictable output
Continuous innovation
Increased availability and resilience of systems

Figure 4. Conceptual model of DevOps capabilities resulting from the systematic literature review.

The second set of capabilities are the *Dynamic DevOps team capabilities*. They represent the processes, structures and systems which constitute the three classes of dynamic capabilities within DevOps teams. The last set of capabilities are capabilities which result from the successful implementation of the other two sets of capabilities. We therefore name this set of capabilities the *Outcome capabilities*. The complete overview of all capabilities and the relationship between these is depicted in Figure 4. The sets of capabilities and their respective literature references will be discussed in the following.

A. *Organizational enabler capabilities for DevOps*

Organizational enabler capabilities serve as a prerequisite for the successful deployment of DevOps teams. They primarily concern the governance of DevOps teams and need to be developed and fostered on an organizational level.

Firstly, DevOps teams require an *organizational vision and strategy* which provides direction and rationale for the DevOps way of working. The management vision is important for addressing barriers during the implementation [34] and ensures alignment of team goals with organizational objectives [35], [36]. One mechanism to achieve this is to deploy appropriate frameworks that provide direction to the DevOps teams [36], [37]. On the other hand, a lack of management support has been identified to be a key challenge in adopting and executing DevOps practices [38], [39].

Since communication and alignment on goals and practices are important elements of DevOps, the *organizational structure* can affect the way of working substantially [19]. It has been suggested that an effective construction for DevOps organizations is to adopt a matrix structure since this setup promotes communication among employees [40]. Other case studies reported that organizations simply merged development and operations departments and introduced new titles such as “DevOps engineer” [16]

Multiple scholars addressed the importance of *leadership commitment and support* towards the DevOps way of working [41]–[43]. Sponsorship from leadership is important during the implementation phase in order to support the organizational culture change [41], [43]. However, continuing leadership commitment remains crucial for existing DevOps teams since managers need to employ collaborative leadership styles instead of trying to control the teams [42]. Leaders should demonstrate lean leadership behavior [40] and practice agile methods themselves rather than just supporting them [42]. We have summarized this evidence under the term *“transformational leadership”*.

Effective DevOps teams often have the decision-making authority to bring software into production whenever desired [9], [44]–[46]. This implies that *team autonomy and decentralized decision-making* should not only be respected and supported by management as discussed earlier but also need to be safeguarded within the role definition of the teams. Furthermore, many scholars stress that DevOps teams should be assigned *shared end-to-end responsibilities* for the entire system life cycle [16], [43], [44], [47]. Employee *incentives and appraisal* processes therefore need to be team-based and promote shared goals [34], [42], [45].

Simultaneous to maintaining the team autonomy, organizations need to foster alignment between teams and ensure that they *share the same processes and* goals [19], [42], [44], [48]. Lastly, organizations need to ensure that DevOps team members receive adequate *training* to fulfill their responsibilities [42] as well as *select and provide them with suitable tooling* [45], [48], [49].

B. *Dynamic DevOps team capabilities*

In this section we present the capabilities that were found to be directly relevant to DevOps teams and enable them to exhibit the three classes of dynamic capabilities. It is important to note that some capabilities may contribute to more than just one class of dynamic capabilities. We therefore present each capability in the context of the class to which we found it to contribute to most dominantly. Furthermore, we found that DevOps dynamic capabilities manifest themselves on two levels. On one hand, *business-related capabilities* relate to the structures, processes and behavior which the teams employ. *Technology-related capabilities* on the other hand concern the technical implementation and automation of systems and processes.

1) Sensing capabilities

Sensing capabilities allow the DevOps teams to sense opportunities and threats regarding their products and services. The first capability which was encountered to support this is the integration of *customer centricity* in the design of processes [36], [42]. DevOps teams should be cross-functional [40], [50] and organized around products- and services [18], [45], [48] instead of individual components. This allows the teams to cover all aspects of the software development life cycle and understand potential shortcomings and feedback.

Another capability which relates to sensing opportunities and threats is the deployment of *intrapreneurship behavior* which is used to describe groups in existing organizations that behave like entrepreneurs [9]. This requires DevOps teams to be skilled at solving problems [2], [51] as well as being assertive and continuously scouting for new opportunities [52].

DevOps teams need to be able to manage and balance expectations from customers as well as stakeholders in their organization. They should therefore adopt *stakeholder management* practices, for example by building relationships with customers [47] and letting them participate in their development process [50], [53], [54].

The last capability which we found to enable the identification of opportunities and threats is *end-user monitoring and feedback*. This requires strong technological skills since teams should aim to extract information regarding end-user behavior from the application and offer users the opportunity to provide feedback on working versions of their product. In order to achieve this. Teams should define suitable metrics and measurements [1], [18], [19], [35], [39] as well as build the capability to aggregate monitoring data into insights [17].

2) Seizing capabilities

Capabilities which allow DevOps teams to seize opportunities target at enabling fast and informed decision-making based on the information which they have received through their sensing capabilities. The prerequisite for doing so is the organizational facilitation of team autonomy which we have discussed earlier in this paper. Lengthy approval processes will hinder the execution of this set of capabilities.

Most dominantly, scholars stress the importance of an *open team culture* [44], [50], [55]. While this capability contributes to all three classes of dynamic capabilities, culture primarily facilitates a continuous exchange of information which is relevant to making decisions. In order to achieve this, team members should trust and respect each other [19], [42] and feel free to share their opinion [51]. Other important elements of this capability are a value-creation oriented mindset [48] and the aim for continuous improvement [1], [42] which ensures that the correct decisions are taken.

Another capability which allows fast decision making is *lean process management* [16], [37] by streamlining and standardizing delivery and approval processes and making progress visible and transparent [46]. Teams should be *self-empowered* which means that they are self-organizing [1], assume responsibility and ownership of their system [45], [56] and continuously experiment to improve current processes. [37]. Furthermore, teams need to independently *align within the team as well as with other teams* when taking decisions to ensure their validity [1], [50], as well as have a clear process for *managing backlog requirements* [37], [46], [53] in order to identify priorities quickly.

The design of product architectures is important to seizing opportunities since they define the manner in which value is delivered to customers [12]. The technological capabilities of *architecture management* [48], [50] as well as *infrastructure management* [43], [57] are therefore crucial to DevOps teams since they determine the boundary conditions within which strategic decisions can be made. Lastly, DevOps teams need to manage their *systems access* by balancing the need to access the production server and deploy on demand [17] while also maintaining the security of the platform and preventing unauthorized access [58], [59].

3) Managing Threats & Transforming

The last class of dynamic capabilities enables DevOps teams to transform their assets following the decisions which they have taken through their seizing capabilities. On a business-level this implies transformation and enhancement of processes, structures and skills within the team. On a technology-level transforming capabilities enable the team to change their system and product.

Transforming assets calls for a clear *change management* process in order to execute change requests quickly [42]. Furthermore, the change process should take care that operations are not affected adversely when deploying a change by aligning responsible actors with each other [42]. A second capability which is relevant to transforming assets are *knowledge management* practices [1], [35] which ensure that all team members have access to available information and can enhance their skill sets. This requires documentation and dissemination of knowledge through appropriate platforms [45], [56] but is also enabled through explicit information sharing and peer reviews [1], [34], [60].

The above mentioned processes can be supported by *Agile or lean project management methodologies* such as Scrum or Kanban. These are deemed to be particularly useful for DevOps teams [9], [42], [52], [55] and can be helpful in quickly transforming assets due to their aim for short throughput times. Agile meeting structures allow for closer communication between team members and regular tracking of goals and progress [49]. Additionally, iterative processes like those proposed in the Scrum framework support frequent knowledge sharing [1] and continuous learning cycles [52]. Lastly, Agile meeting structures such as sprint planning events and standups [37] can support the *continuous planning* capabilities [19], [54] which DevOps teams need in order to adjust their planning based on new insights.

From a technology perspective, teams need to be able to adapt their systems or services on demand. *Continuous software engineering* practices like continuous integration, continuous testing and continuous deployment [53], [60], [61] as well as the automation of versioning [16], [46] and recovery activities [19], [43] allow teams to bring software changes quickly into production. This aim is enabled by the effective *management of artifacts* [39], [55] and *management of configurations* of software and infrastructure [47], [59], [62]. For example, configuration management files may be stored and versioned in repositories where they are triggered by the team whenever needed [55]. Next to these development and deployment capabilities, DevOps teams should develop *operations management* capabilities [54], [63] in order to keep their systems running smoothly. This includes handling incident response activities [39] and automating the restart of environments in case of instabilities [43].

While the previously discussed capabilities are primarily aimed at transforming assets, we also found two capabilities to be directly relevant to managing threats. Firstly, DevOps teams need to *manage security* [38], [58], [59] by analyzing security requirements [59], [62] and integrating security testing into their process [58]. Furthermore, we found that the use of continuous and automated *monitoring and logging techniques* is important to managing risks since they ensure traceability and compliance of systems and infrastructure [16], [45], [52].

C. *DevOps outcome capabilities*

The last set of capabilities are those that were identified to be a result of the successful implementation of the previously discussed capabilities. In academic literature, these outcomes are often described as "benefits" of DevOps. However, the outcomes were often reported on the level of perception by the organizations employing DevOps but were rarely measured based on objective metrics.

Perera, Bandara, and Perera [34] argue that there is strong evidence that DevOps improves the perceived *agility* of organizations. They define agility in terms of an organizations adaptability to new technology which is slightly different to the definition by Nagarajan and Overbeek [42] who view this concept as the ability to respond fast to customer and market changes. Besides this, multiple authors report that DevOps practices led to the perceived improvement of *quality assurance* [44], [55]. This outcome is regarded to be a result of the combination of various DevOps practices such as automated testing, versioning [44] and monitoring [61].

In a similar manner, companies using DevOps reported to have improved their *software deployment frequency* [41], [63] and felt that they created and delivered more *value* to their customers [41], [48]. This is partially due to the automation of processes but also because DevOps capabilities enable a better identification of business needs [42]. Furthermore, the standardization and automation of processes and frameworks led to *higher predictability* of quality and frequency of the output [37] and *simplification of processes and documentation* [43], [48].

DevOps also was reported to increase the *innovative output* [9] of the teams due to the cyclical way of working. Lastly, systems were reported to be more *stable* and had a higher *availability* [43], [44] which is in line with our previous findings on quality assurance.

V. Discussion and research agenda

The aim of this paper was to develop a conceptual model that explains how DevOps teams can contribute to firm competitive advantage. We have taken the novel approach of framing the insights gained from a systematic literature review on DevOps capabilities in the context of the established theory of dynamic capabilities [11], [22]. The resulting framework indicates that DevOps teams can contribute to competitive advantage by building sensing, seizing and transforming capabilities both on a business-level as well as on a technology-level.

Our research rationale supports and extends the current body of knowledge related to DevOps capabilities such as the work of Teixeira, Pereira, Henriques, Silve, & Faustino [7] who have conducted a systematic review on DevOps capabilities and areas. Furthermore, Badshah, Khan, and Khan [6] have conducted a systematic review on DevOps maturity models and capabilities. Sánchez-Gordón and Colomo-Palacios [5] performed a review on DevOps culture attributes and provide an overview of cultural elements which equally supports our findings. Wiedemann and Schulz [9] have used the resource-based lens to demonstrate the effect of DevOps capabilities on software process innovation. Our review extends the aforementioned works by providing a comprehensive overview of business and IT capabilities and additionally offering a conceptual model with an explanatory argumentation how these capabilities can benefit DevOps organizations, based on an established theory of strategic advantage. To the best of our knowledge, the research at hand is the first research to do so.

Extant literature has debated whether dynamic capabilities have a direct influence on firm competitive advantage as hypothesized by Teece, Pisano, and Shuen [11] or whether this link is indirect and dependent on factors like firm strategy as proposed by Wang and Ahmed [64] or Wade and Hulland [24]. We can therefore not conclude that the identified DevOps capabilities have a direct influence on the strategic success of IT organizations but we do argue that they provide a valuable contribution to firm competitive advantage nevertheless. Furthermore, it is important to emphasize that the aforementioned argumentation has not been empirically proven and therefore remains on a conceptual level.

Additionally, Teece, Pisano, and Shuen [11] as well as Eisenhardt and Martin [22] argue that dynamic capabilities are idiosyncratic in their details. This constitutes the very nature of dynamic capabilities since they need to be hard to imitate in order to be of strategic value to the company employing them [11]. As a consequence of this, we can identify what the relevant dynamic capabilities are on a high level but we cannot list in detail how a company should implement them exactly. While we have presented some examples, every company is expected to implement the capabilities which we identified in their own unique way.

This paper has introduced the dynamic capabilities theory as an enabler to the more recent concept of enterprise agility which is defined as *"the ability of firms to sense environmental change and respond readily"* [14]. We therefore argue that the capabilities discussed in this paper are equally relevant to organizations seeking to enhance their organizational agility as to those who are seeking for competitive advantage by means of dynamic capabilities.

A. *Contributions to research and practice*

Our findings have implications for both research and practice. On the theoretical side, our research provides novel contributions to the body of knowledge in information systems literature as well as strategic management literature. We present a conceptual model on how the DevOps approach is of strategic value and contributes to firm competitive advantage. The aim of this model is to generate a deeper understanding of the impact which the implementation of agile software development approaches can have in IT organizations. Simultaneously, this research demonstrates the relevance and application of the dynamic capabilities theory in the age of digital transformation.

On the practical side, we present insights for practitioners seeking to implement the DevOps approach successfully. By separating the DevOps team dynamic capabilities from enabler capabilities and outcome capabilities, we argue that organizations should prioritize the development of enabling capabilities before moving towards the other two sets of capabilities. We propose that the strategic success of DevOps teams is highly contingent on this form of organizational support. Factors like organizational structure, leadership support and the embedding of decentralized-decision making are key components to unlocking DevOps team dynamic capabilities. Lastly, we provide a comprehensive list of dynamic capabilities on which emerging DevOps teams can focus.

B. *Limitations & Future Work*

The limitations of this research are primarily grounded in the design choices which were made during the execution phase of the systematic literature review. We have chosen to exclude papers from the review that were not empirically grounded or empirically validated which led to a relatively small subsample of eligible papers. A second review could occur in the future taking into account both empirical as well as conceptual papers on DevOps capabilities to extend the results of the current paper.

As part of our literature synthesis, we have employed elements from the grounded theory approach. Although we have executed and documented the process diligently, it is possible that previous knowledge has implicitly influenced our definition of codes and code categories [31]. In future research, the results should therefore be verified by other scholars.

We advocate that future research on DevOps capabilities should incorporate two considerations. First, whereas the current body of literature has already explored the fundamental nature of DevOps capabilities, it is not yet fully understood how the idiosyncratic configuration of these capabilities is shaped and influenced. We therefore call for more research that explores the interaction and effect of DevOps capabilities with their organizational environment. The conceptual model presented in this paper constitutes a first step in this effort. Second, we found that the outcomes of DevOps were often reported in terms of perceived benefits but were rarely measured using objective metrics. We therefore believe that more research is necessary to verify both operational as well as strategic outcomes of DevOps.

VI. Conclusion

The aim of this paper was to develop a conceptual model on how the DevOps approach contributes to firm competitive advantage. To this end we have conducted a systematic literature review of 37 papers on DevOps capabilities. We have analyzed our results in the light of the dynamic capabilities theory [12] and propose that DevOps teams can indeed build dynamic capabilities which allow them so sense opportunities and threats, seize opportunities and transform their assets. These capabilities need to be built on a business-level as well as on a technology-level. We furthermore propose that the success of DevOps teams is highly contingent on the presence of a set of organizational enabler capabilities. Given the successful implementation of both sets of capabilities, DevOps organizations can expect a set of beneficial outcome capabilities.

Our research rationale is grounded in existing literature on enterprise agility, dynamic capabilities and DevOps. We extend this body of knowledge by providing a detailed overview of DevOps capabilities and a conceptual model on the impact of DevOps capabilities on firm competitive advantage. The research has implications for both theory and practice.

Acknowledgments

We would like to thank Quint Nederland B.V. for their support and interest over the course of this research. We would also like to thank the anonymous reviewers for their constructive comments.

References

[1] A. Hemon, B. Fitzgerald, B. Lyonnet, and F. Rowe, “Innovative practices for knowledge sharing in large-scale DevOps,” *IEEE Softw.*, vol. 37, no. 3, pp. 30–37, 2020.

[2] A. Wiedemann, M. Wiesche, and H. Krcmar, “Integrating development and operations in cross-functional teams - Toward a DevOps competency model,” in *Proceedings of the 2019 Computers and People Research Conference*, 2019, pp. 14–19.

[3] B. Fitzgerald and K. J. Stol, “Continuous software engineering: A roadmap and agenda,” *J. Syst. Softw.*, vol. 123, pp. 176–189, 2017.

[4] I. M. Sebastian, J. W. Ross, C. Beath, M. Mocker, K. G. Moloney, and N. O. Fonstad, “How big old companies navigate digital transformation,” *MIS Q. Exec.*, vol. 16, no. 3, pp. 197–213, 2017.

[5] M. Sánchez-Gordón and R. Colomo-Palacios, “Characterizing DevOps culture: A systematic literature review,” in *Software Process Improvement and Capability Determination. SPICE 2018. Communications in Computer and Information Science*, 2018, pp. 3–15.

[6] S. Badshah, A. A. Khan, and B. Khan, “Towards process improvement in DevOps: A systematic literature review,” in *EASE '20: Proceedings of the Evaluation and Assessment in Software Engineering*, 2020, pp. 427–433.

[7] D. Teixeira, R. Pereira, T. A. Henriques, M. Silva, and J. Faustino, “A systematic literature review on DevOps capabilities and areas,” *Int. J. Hum. Cap. Inf. Technol. Prof.*, vol. 11, no. 3, pp. 1–22, 2020.

[8] N. Forsgren and J. Humble, “The role of continuous delivery in IT and organizational performance,” *SSRN Electron. J.*, pp. 1–15, 2015.

[9] A. Wiedemann and T. Schulz, “Key capabilities of devops teams and their influence on software process innovation: A resource-based view,” in *Proceedings of the 23rd Americas Conference on Information Systems, AMCIS 2017*, 2017.

[10] T. Ravichandran, “Exploring the relationships between IT competence, innovation capacity and organizational agility,” *J. Strateg. Inf. Syst.*, vol. 27, no. 1, pp. 22–42, 2018.

[11] D. J. Teece, G. Pisano, and A. Shuen, “Dynamic capabilities and strategic management,” *Strateg. Manag. J.*, vol. 18, no. 7, pp. 509–533, Aug. 1997.

[12] D. J. Teece, “Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance,” *Strateg. Manag. J.*, vol. 28, no. 13, pp. 1319–1350, 2007.

[13] D. Teece, M. Peteraf, and S. Leih, “Dynamic capabilities and organizational agility: Risk, uncertainty, and strategy in the innovation economy,” *Calif. Manage. Rev.*, vol. 58, no. 4, pp. 13–35, 2016.

[14] E. Overby, A. Bharadwaj, and V. Sambamurthy, “Enterprise agility and the enabling role of information technology,” *Eur. J. Inf. Syst.*, vol. 15, no. 2, pp. 120–131, 2006.

[15] J. Webster and R. T. Watson, “Analyzing the past to prepare for the future: Writing a literature review.,” *MIS Q.*, vol. 26, no. 2, pp. xiii–xxiii, 2002.

[16] F. M. A. Erich, C. Amrit, and M. Daneva, “A qualitative study of DevOps usage in practice,” *J. Softw. Evol. Process*, vol. 29, no. 6,

2017.

[17] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "An exploratory study of DevOps: Extending the dimensions of DevOps with practices," in *ICSEA 2016 : The Eleventh International Conference on Software Engineering Advances*, 2016, pp. 91–99.

[18] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "Dimensions of DevOps," in *Agile Processes in Software Engineering and Extreme Programming. XP 2015. Lecture Notes in Business Information Processing*, 2015, pp. 212–217.

[19] J. Smeds, K. Nybom, and I. Porres, "DevOps: A definition and perceived adoption impediments," in *Agile Processes in Software Engineering and Extreme Programming. XP 2015. Lecture Notes in Business Information Processing*, 2015, pp. 166–177.

[20] J. Humble and J. Molesky, "Why enterprises must adopt devops to enable continuous delivery," *Cut. IT J.*, vol. 24, no. 8, pp. 6–12, 2011.

[21] A. Wiedemann, "IT governance mechanisms for DevOps teams - How incumbent companies achieve competitive advantages," in *Proceedings of the 51st Hawaii International Conference on System Sciences, HICSS 2018*, 2018, pp. 4931–4940.

[22] K. M. Eisenhardt and J. A. Martin, "Dynamic capabilities: What are they?," *Strateg. Manag. J.*, vol. 21, no. 10/11, pp. 1105–1121, 2000.

[23] C. Brosig, M. Westner, and S. Strahringer, "Revisiting the concept of IT capabilities in the era of digitalization," in *2020 IEEE 22nd Conference on Business Informatics (CBI)*, 2020, pp. 84–93.

[24] M. Wade and J. Hulland, "Review: The resource-based view and information systems research: Review, extension, and suggestions for future research," *MIS Q.*, vol. 28, no. 1, pp. 107–142, 2004.

[25] B. D. Rouhani, M. N. Mahrin, F. Nikpay, R. B. Ahmad, and P. Nikfard, "A systematic literature review on enterprise architecture implementation methodologies," *Inf. Softw. Technol.*, vol. 62, no. 1, pp. 1–20, Jun. 2015.

[26] B. Kitchenham, "Procedures for performing systematic reviews," 2004.

[27] J. Vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, and A. Cleven, "Reconstructing the giant: On the importance of rigour in documenting the literature search process," in *17th European Conference on Information Systems, ECIS 2009*, 2009.

[28] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, and The PRISMA Group, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *PLoS Med.*, vol. 6, no. 7, pp. 1–6, Jul. 2009.

[29] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*, 2014, pp. 1–10.

[30] A. A. Booth *et al.*, "Guidance on choosing qualitative evidence synthesis methods for use in health technology assessments of complex interventions," 2016. [Online]. Available: http://www.integrate-hta.eu/downloads/.

[31] J. F. Wolfswinkel, E. Furtmueller, and C. P. M. Wilderom, "Using grounded theory as a method for rigorously reviewing literature," *European Journal of Information Systems*, vol. 22, no. 1. Palgrave Macmillan Ltd., pp. 45–55, 2013.

[32] M. Dixon-Woods, "Using framework-based synthesis for conducting reviews of qualitative studies," *BMC Med.*, vol. 9, no. 39, 2011.

[33] C. Carroll, A. Booth, and K. Cooper, "A worked example of 'best fit' framework synthesis: A systematic review of views concerning the taking of some potential chemopreventive agents," *BMC Med. Res. Methodol.*, vol. 11, no. 29, 2011.

[34] P. Perera, M. Bandara, and I. Perera, "Evaluating the impact of DevOps practice in Sri Lankan software development organizations," in *16th International Conference on Advances in ICT for Emerging Regions, ICTer 2016*, 2016, pp. 281–287.

[35] S. W. Hussaini, "A systemic approach to re-inforce development and operations functions in delivering an organizational program," in *Procedia Computer Science*, 2015, vol. 61, pp. 261–266.

[36] S. W. Hussaini, "Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 178–183.

[37] R. K. Gupta, M. Venkatachalapathy, and F. K. Jeberla, "Challenges in adopting continuous delivery and DevOps in a globally distributed product team: A case study of a healthcare organization," in *2019 ACM/IEEE 14th International Conference on Global Software Engineering, ICGSE 2019*, 2019, pp. 30–34.

[38] N. Tomas, J. Li, and H. Huang, "An empirical study on culture, automation, measurement, and sharing of DevSecOps," in *2019 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2019*, 2019.

[39] M. A. Akbar *et al.*, "Prioritization based taxonomy of DevOps challenges using fuzzy AHP analysis," *IEEE Access*, vol. 8, pp. 202487–202507, 2020.

[40] M. A. Akbar, S. Mahmood, M. Shafiq, A. Alsanad, A. A.-A. Alsanad, and A. Gumaei, "Identification and prioritization of DevOps success factors using fuzzy-AHP approach," *Soft Comput.*, 2020.

[41] L. Riungu-Kalliosaari, S. Mäkinen, L. E. Lwakatare, J. Tiihonen, and T. Männistö, "DevOps adoption benefits and challenges in practice: A case study," in *Product-Focused Software Process Improvement. PROFES 2016. Lecture Notes in Computer Science*, 2016, pp. 590–597.

[42] A. D. Nagarajan and S. J. Overbeek, "A DevOps implementation framework for large Agile-based financial organizations," in *On the Move to Meaningful Internet Systems. OTM 2018 Conferences. OTM 2018. Lecture Notes in Computer Science*, 2018, pp. 172–188.

[43] W. P. Luz, G. Pinto, and R. Bonifácio, "Adopting DevOps in the real world: A theory, a model, and a case study," *J. Syst. Softw.*, vol. 157, 2019.

[44] W. P. Luz, G. Pinto, and R. Bonifácio, "Building a collaborative culture: A grounded theory of well succeeded DevOps adoption in practice," in *ESEM '18: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2018, pp. 1–10.

[45] M. Senapathi, J. Buchan, and H. Osman, "DevOps capabilities, practices, and challenges: Insights from a case study," in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, 2018, pp. 57–67.

[46] S. Al-Zahrani and B. Fakieh, "How devops practices support digital transformation," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 2780–2788, 2020.

[47] W. Hussain, T. Clear, and S. MacDonell, "Emerging trends for global DevOps: A New Zealand perspective," in *Proceedings - 2017 IEEE 12th International Conference on Global Software Engineering, ICGSE 2017*, 2017, pp. 21–30.

[48] C. Crowley, L. McQuillan, and C. O'Brien, "Understanding DevOps: Exploring the origins, composition, merits, and perils of a DevOps Capability," in *Proceedings of the 4th International Conference on Production Economics and Project Evaluation, ICOPEV 2018, Guimarães, Portugal*, 2018, pp. 29–37.

[49] A. A. Khan and M. Shameem, "Multicriteria decision-making taxonomy for DevOps challenging factors using analytical hierarchy process," *J. Softw. Evol. Process*, vol. 32, no. 10, Oct. 2020.

[50] R. de Feijter, S. Overbeek, R. van Vliet, E. Jagroep, and S. Brinkkemper, "DevOps competences and maturity for software producing organizations," in *Enterprise, Business-Process and Information Systems Modeling. BPMDS 2018, EMMSAD 2018. Lecture Notes in Business Information Processing*, 2018, pp. 244–259.

[51] F. Erich, "Devops is simply interaction between development and operations," in *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. DEVOPS 2018. Lecture Notes in Computer Science*, 2019, pp. 89–99.

[52] R. Srinivasan, S. D. Eppinger, and N. Joglekar, "The structure of Devops in product-service system development," in *Proceedings of the International Conference on Engineering Design, ICED*, 2019, pp. 3111–3120.

[53] V. Gruhn and C. Schäfer, "BizDevOps: Because DevOps is not the end of the story," in *Intelligent Software Methodologies, Tools and Techniques. SoMeT 2015. Communications in Computer and Information Science*, 2015, pp. 388–398.

[54] M. Zaydi and B. Nassereddine, "DevSecOps practices for an agile

and secure IT service management," *J. Manag. Inf. Decis. Sci.*, vol. 23, no. 2, pp. 134–149, 2020.

[55] L. E. Lwakatare *et al.*, "DevOps in practice: A multiple case study of five companies," *Inf. Softw. Technol.*, vol. 114, pp. 217–230, 2019.

[56] A. Wiedemann, "A new form of collaboration in IT teams – Exploring the DevOps phenomenon," in *Proceedings ot the 21st Pacific Asia Conference on Information Systems, PACIS 2017*, 2017.

[57] J. Díaz, R. Almaraz, J. Pérez, and J. Garbajosa, "DevOps in practice - an exploratory case study," in *XP '18: Proceedings of the 19th International Conference on Agile Software Development: Companion*, 2018.

[58] V. Mohan, L. ben Othmane, and A. Kres, "BP: Security concerns and best practices for automation of software deployment processes -An industrial case study," *2018 IEEE Secur. Dev. Conf.*, pp. 21–28, 2018.

[59] A. Rahman and L. Williams, "Security practices in DevOps," in *HotSos '16: Proceedings of the Symposium and Bootcamp on the Science of Security*, 2016, pp. 109–111.

[60] R. W. Macarthy and J. M. Bass, "An empirical taxonomy of DevOps in practice," in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2020, pp. 221–228.

[61] V. Gupta, P. K. Kapur, and D. Kumar, "Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling," *Inf. Softw. Technol.*, vol. 92, pp. 75–91, 2017.

[62] A. Rahman and L. Williams, "Software security in DevOps: Synthesizing practitioners' perceptions and practices," in *Proceedings - International Workshop on Continuous Software Evolution and Delivery, CSED 2016*, 2016, pp. 70–76.

[63] J. Díaz, J. E. Perez, A. Yague, A. Villegas, and A. de Antona, "DevOps in practice – A preliminary analysis of two multinational companies," in *Product-Focused Software Process Improvement. PROFES 2019. Lecture Notes in Computer Science*, 2019, pp. 323–330.

[64] C. L. Wang and P. K. Ahmed, "Dynamic capabilities: A review and research agenda," *International Journal of Management Reviews*, vol. 9, no. 1. pp. 31–51, Mar-2007.