Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute
InformingScience.org

IJIKM.org

Volume 15, 2020

AN EXPLORATORY STUDY ON THE DEVOPS IT ALIGNMENT MODEL

Michael Hart* College of Science, Engineering, & Technology, Minnesota State University, Mankato, MN, USA michael.hart-2@mnsu.edu

John Burke College of Science, Engineering, & Technology, Minnesota State University, Mankato, MN, USA john.burke@mnsu.edu

ABSTRACT

Aim/Purpose Based on business-IT alignment, this study addresses the understudied practice of DevOps.

Background Although organizations continue to implement DevOps practices, few studies explore connections with prior theory. This study contributes to this need by developing the DevOps strategic IT alignment model.

Methodology The sample included 57 firms from the current Forbes Global 2000 and the Fortune 500 lists. The authors employed partial least squares structural equation modeling (PLS-SEM) to evaluate the DevOps IT alignment model.

Contribution The proposed model builds a foundation for further investigation into the influence of theory on DevOps using quantitative research methods. It also contributes to a reliable and valid DevOps instrument for future exploration.

Findings Continuous integration of software and knowledge sharing increases the level of IT subunit alignment in large organizations that foster DevOps. Furthermore, practicing DevOps positively influences the level of business-IT alignment.

Recommendations for Practitioners Organizations that cultivate DevOps experience greater levels of business-IT alignment through stronger knowledge sharing and continuous integration of applications. Thus, managers should identify how to develop closer bonds between subunits with dissimilar skillsets in their organizations.

Recommendations for Researchers Researchers should explore how theories interact, help, and/or do not support blossoming practices like DevOps.

Accepting Editor Iris A Humala │ Received: April 17, 2020│ Revised: June 15, June 24, 2020 │ Accepted: June 26, 2020.
Cite as: Hart, M., & Burke. J. (2020). An exploratory study on the DevOps IT alignment model. *Interdisciplinary Journal of Information, Knowledge, and Management*, *15,* 127-154. https://doi.org/10.28945/4595

(CC BY-NC 4.0) This article is licensed to you under a Creative Commons Attribution-NonCommercial 4.0 International License. When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

Impact on Society	Stronger bonds increase knowledge sharing between interdepartmental colleagues. Lower hierarchical levels of an organization as well as higher managerial levels benefit from cross-domain IT knowledge.
Future Research	It is important to explore how different types of knowledge in diverse disciplines requires unique cross-discipline bonds to form and whether these relationships have connections with the contingency theory and quality management.

Keywords	DevOps, strategic IT alignment, contingency theory, continuous integration, knowledge sharing, continuous deployment, software theory

INTRODUCTION

Although organizations are implementing DevOps practices, few studies in literature explore connections with prior theory (Erich et al., 2017; Jabbari et al., 2016). The goal of this study is to investigate whether two pillars of DevOps, namely, continuous integration and knowledge sharing between the IT operations and IT development departments, associate with the relational and structural dimensions of business-IT alignment. Business-IT alignment, with its rich history of measuring IT performance (Chan et al., 2006; Chan & Reich, 2007; Chen et al., 2008; Kearns & Lederer, 2003; Kearns & Sabherwal, 2007; Luftman & Brier, 1999; Preston & Karahanna, 2009; Wang et al., 2015), provides a potential stepping-stone for discovering the deeper connections of DevOps with existing theory. Literature suggests that the contingency theory viewpoint of quality management can enable strategic alignment when environmental business conditions become more complex and require a greater degree of malleability (McAdam et al., 2019). Within the IT department of organizations with the latter environmental conditions, DevOps is capable of scaling quickly based upon changing profitability and fluctuations in the underlying technologies necessary for success (Fokaefs et al., 2017).

The practice of IT development and IT operations collaborating through the concept referred to as "DevOps" distinguishes the goal of achieving faster software releases to production environments without compromising the quality of the solution (Gupta et al., 2017). Similar to a core dimension of its success, swift adaptability, the conception of DevOps is in a continual state of evolution due in part to its infant state. DevOps is a positive step toward attaining a leaner systems development life cycle by bringing software development, quality assurance, IT operations subunits and several other engineering personnel closer together. Organizations choose to adopt DevOps not only to reduce software delivery times, but also to improve customer feedback and problem-solving capabilities between employees (Erich et al., 2017). DevOps champions cross-functional collaboration, continuous integration of code, knowledge sharing, testing, rapid deployment of software, and the monitoring of this cycle without negatively influencing the quality of the application (Erich et al., 2014a; Gupta et al., 2017). Software development benefits from DevOps not merely due to automation, agile methods, and tools but from a cultural shift. More specifically, an environment conducive to collaboration is necessary between traditionally isolated IT subunits (Erich et al., 2014a, 2014b, 2017).

While DevOps as a discipline is still maturing, initial attempts to develop a formal definition include concepts such as continuous collaboration, communication, integration of software, quality assurance, and automated deployment of software (Erich et al., 2017; Jabbari et al., 2016). Immediate achievement refers to the "continuous" attribute of each DevOps component in the author's work. It also has ties to the progression of modern digital business applications in coordination with advancements like cloud computing (Airaj, 2016; Fokaefs et al., 2017; Wettinger et al., 2016). Despite the initial scholarly progress of this young practice, extensive reviews of relevant literature have noted future work needs to explore whether DevOps is effective using quantitative research (Erich et al., 2017). There are also key aspects of DevOps that early studies identified yet are not emphasized as often in current literature such as how knowledge is shared between departments that do not understand each other's discipline well (Colomo-Palacios et al., 2018; Humble & Molesky, 2011). This study expands DevOps research by using quantitative methods to explore its effectiveness in correlation with a richly studied concept in information systems, the strategic alignment of business and IT. Furthermore, results add validation to the DevOps IT alignment model (Hart, 2017).

The organization of this work will be as follows. It begins by introducing the background of DevOps in current literature. Next, the researchers propose an extended DevOps theoretical model. Hypotheses to test this model proceed thereafter. The research design outlines the sample and research methods. Evaluation of the DevOps IT alignment model proceeds thereafter. Finally, the authors discuss the meaning of the results and conclude the study.

BACKGROUND

Prior to investigating connections to theory, it is important to understand the underlying concepts that define DevOps. Leite et al. (2020), in a wide-reaching study of DevOps challenges, agree with prior extensive literature reviews (e.g., Jabbari et al., 2016), concluding that DevOps needs an accepted definition. A second objective is to investigate the value of DevOps. The authors will explore performance measures specifically. Despite the conclusion of comprehensive reviews of DevOps literature that a volume of DevOps results insufficiently validates positive IT operations performance (Erich et al., 2014a), organizations are continuing to invest in DevOps due to its perceived value (Erich et al., 2017; Gupta et al., 2017). Third, the researchers will examine connections of DevOps with existent theory, which currently is not present in literature (Erich et al., 2017; Jabbari et al., 2016). Its infant state is a natural reason a deep foundation in theory is not present (see Erich et al., 2014a).

A more standardized definition of DevOps will help guide future literature and assist in the exploration of connections with theory (Jabbari et al., 2016). Thus, a primary goal of the review of literature is to investigate the evolving definition of DevOps. Subsequently, the authors will review literature that studies DevOps and its connection to theory.

EARLY EFFORTS TO DEVELOP A COMMON DEFINITION OF DEVOPS

This section outlines a logical progression of concepts that shape the definition of DevOps in literature. This includes DevOps in the context of agile methods, cloud computing, organizational adoption, and prior literature.

DevOps often supports or compliments cloud computing and agile software development strategies. To be competitive, organizations must deliver customer pleasing prototypes in iterative and rapid cycles. Like DevOps, agile life cycles and cloud computing have found their stride because they help organizations compete by delivering more value to their customers at lower price points (Airaj, 2016; Fokaefs et al., 2017; Jabbari et al., 2016; Li et al., 2011; Liu et al., 2014).

DevOps defined in context of agile methodologies

One of the prominent themes in the early stages of defining DevOps included agile methodologies. The process of software development at many organizations was linear prior to this shift. For instance, when the software development unit concluded a block of code at the end of a cycle, quality assurance (QA) took over the process. Support had to increase to decrease release cycles. There were predicable patterns of workload and looping processes in this era of development, however, strict adherence to sequences slowed software cycles. Agile software development life cycles have eased the associated delays (Roche, 2013).

While DevOps does not meet all the objectives of the agile manifesto nor was this an original goal, one of the earliest published case studies at IBM validated its ability to deliver software to customers at an unparalleled rate. It achieved this through cross team collaboration and continuous integration and automation of software (Liu et al., 2014). Agile and DevOps both focus on shorter product release cycles. However, DevOps is distinct from agile methods through automating software analysis, regular monitoring, and precise procedures such as infrastructure as code (Jabbari et al., 2016).

DevOps defined in context of cloud computing

Certain studies also closely correlate the progressions of modern technology with the concepts of cloud computing and DevOps (Airaj, 2016; Fokaefs et al., 2017; Wettinger et al., 2016)). Fokaefs et al. (2017) state that the two primary influences of electronic business and application technologies are DevOps and cloud computing (p. 1). To be effective, results indicated integration of DevOps and business operations (BizOps) is necessary to automate scaling based upon revenue (Fokaefs et al., 2017). Cloud computing is viewed as an effective platform for implementing DevOps automation and tool sets. Similarly, cloud computing is often enhanced by DevOps practices through efficiencies in the automation and integration of reusable artifacts such as code and scripts. Additionally, DevOps and cloud computing assist with the automation of acquiring valuable expert knowledge within an organization (Wettinger et al., 2015).

Although the principle of model-driven cloud management is perhaps a competing strategy to DevOps, it is not a replacement. IT operations and IT development use many different forms of offline shell scripts, software, and tools that DevOps teams integrate extrinsic to model-driven cloud management. While DevOps artifacts are often cloud-based, as a practice, they can be independent of their deployment architecture (Jabbari et al., 2016; Wettinger et al., 2015).

Agile and cloud computing both support and often overlap with DevOps in their purpose to add value to business by delivering high quality software to customers at a faster rate. Nevertheless, they also have several distinctions. These characteristics help shape the definition of DevOps. To understand the latter differences several mapping studies analyzed and assessed a volume of DevOps articles. Erich et al. (2014b) performed a DevOps mapping study on 26 articles. Their analysis formed the labels 1) automation, 2) a culture of collaboration, 3) how DevOps can integrate into structures and standards, 4) sharing knowledge between development and operations, 5) quality assurance, 6) service development, and 7) creating development and operation measurements. Several of these labels emphasize one of the core purposes of DevOps, to unify IT development and operations into one business unit (Erich et al., 2014a). A higher-level perspective perhaps is that DevOps provides a modern framework for integrating IT subunits with distinct knowledge areas.

While agile and cloud computing were early trademarks of DevOps, its practice has expanded into many other areas of IT research. Literature that is more recent focuses identifying the key enablers necessary for its successful adoption.

DevOps defined in context of its adoption

As the popularity of DevOps increases, certain researchers have emphasized the importance of its successful adoption. Prior to adoption, organizations need a more refined definition to understand its integration into existing processes and structures. Smeds et al. (2015) constructed the definition of DevOps as "a set of engineering process capabilities supported by certain cultural and technological enablers. Capabilities define processes that an organization should be able to carry out, while the enablers allow a fluent, flexible, and efficient way of working" (p. 170). The scholars further dichotomize the latter three enablers to help firms increase their capabilities during adoption.

The enabler of capability involves software and service engineering such as continuous planning, development, integration, testing, deployment, and feedback that allows these activities to occur in small additions and with little delay. Small incremental additions with the least amount of delay associate with the word "continuous." Cultural enablers refer to the collaborative environment that allows for excellent communication, shared values, trust, communal ownership, and learning amongst team members. The third and final enabler, technological, focuses on automating tasks. This includes build, test, deployment, and monitoring automation. Holistically, the capability enabler is the centerpiece of DevOps adoption, requiring the support of cultural and technological enablers to be successful (Smeds et al., 2015).

Beyond defining DevOps from the context of its implementation, another popular approach to defining a concept is using mapping studies.

DevOps defined by the central components of existent literature

Jabbari et al. (2016) underline the importance of the seamless flow of knowledge between IT development and IT operations to achieve faster deployments of high-quality software. Their study sought to propose a consistent definition of DevOps by investigating 49 articles. It constructed eight primary components from its exhaustive review of literature, delineated as C1-C8. The mapping found that the definition of DevOps includes the combination of development and operations (C1), communication, collaboration, teamwork (C2), and a set of practices that bridge gaps between the two subunits (C3). Furthermore, it integrates developers and operators (C4), invokes a continuous feedback loop during software delivery (C5), automates and improves software deployment (C6), increases continuous integration (C7), and helps with the quality assurance of software (C8). The resulting definition states, “DevOps is a development methodology (C4) aimed at bridging the gap (C3) between Development (Dev) and Operations (C1), emphasizing communication and collaboration (C2), continuous integration (C7), quality assurance (C8) and delivery (C5) with automated deployment (C6) utilizing a set of development practices” (Jabbari et al., 2016, p. 6).

Leite et al. (2020) proposed a definition based upon those most cited in their review of literature: “DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability” (p. 127:1). The authors argue the definition is sufficient because it represents the less technical and more technical concepts of the practice. Less technical concepts infer multidisciplinary collaboration or the behavioral aspects. Whereas automation and delivery assure the correctness and reliability of software and correlate with the more technical aspects of DevOps (Leite et al., 2020).

Variations in DevOps definitions

The DevOps definitions discussed in this review of literature illustrate variations at the highest conceptual level of the term. For instance, scholars refer to DevOps as a development methodology (Jabbari et al., 2016), a conceptual framework (Erich et al., 2014a), or a set of engineering process capabilities (Smeds et al., 2015). Within the definition language, Smeds et al. (2015) seem to focus on higher-level concepts defined in the scope of engineering capabilities rather than naming specific proficiencies of DevOps within the definition itself. For example, they do not mention software development, collaboration between personnel, quality assurance, deployment, automation, or continuous integration. Leite et al. (2020) take a similar approach by defining DevOps using two primary less technical and more technical components, such as collaboration and continuous delivery. In contrast, Jabbari et al. (2016) include the latter concepts in their definition language. Slight definitional distinctions from the Erich et al. (2014b) and the Jabbari et al. (2016) studies include measurements, service development, and information sharing. Regardless, many of the shared dimensions distinctly identify DevOps in contrast to other strategies, for example, the focus on combining IT subunits and continuous integration of software (Jabbari et al., 2016). Given the variations in the approaches to defining DevOps and the evolution of modern DevOps practices, additional efforts should update the definition of DevOps.

While several components of a central DevOps definition are matriculating, Shahin et al. (2018) reemphasized a core objective that early case studies at IBM based this new concept upon: delivering higher quality software to customers at a faster rate (Liu et al., 2014). Furthermore, in order for several primary objectives to be achieved, such as bridging the so-called gaps between IT development and operations (e.g., gaps in communication, collaboration, teamwork) (Jabbari et al., 2016), early exploratory DevOps literature proposed that the two subunits must be strategically aligned (Liu et al., 2014). Thus, the authors of this study advance to a brief investigation into these foundations as they provide a potential roadmap toward connections with existent theory.

Early Efforts that Connect DevOps with Fundamental IT Theory and Philosophies

Systematic literature reviews and mapping studies have made recent attempts to better define DevOps and its primary effects within an organization. Regardless, as these studies illustrate, there are still few efforts exploring potential theories supporting this practice (Erich et al., 2017; Jabbari et al., 2016; Smeds et al., 2015). A primary anchor of early DevOps research is business-IT alignment (Hart, 2017; Iden et al., 2012).

Business-IT alignment refers to the elements within an organization that attribute to the harmonization of business strategies and IT strategies to help it achieve certain competitive advantages (Luftman & Brier, 1999). Exploratory research also indicates that there is a connection between the IT alignment model and the alignment of IT development with IT operations (Iden et al., 2011, 2012). How an organization aligns its information systems strategy with its business strategy is a foundational concept behind this model. One of the original conceptual models illustrates that as information systems and business strategies are oriented, there is an increase in organizational performance and information systems effectiveness (Chan & Huff, 1993). Early strategic alignment models highlighted four primary dimensions that could increase effectiveness. These included organizational strategy, IT strategy, IT infrastructure, and organizational infrastructure (Henderson, & Venkatraman, 1993). The primary problems identified between these organizational subunits are the lack of knowledge sharing, communication, planning, and partnership (Iden et al., 2012).

Although not all scholars would agree that the contingency theory is indeed a theory, it is one of the cited theories in association with IT alignment (Chan & Reich, 2007). Literature has suggested that the unison of organizational structure and context must exist for a firm to compete (Chan & Reich, 2007; Drazin & Van de Ven, 1985) because, as the contingency theory posits, a specific environment will not necessarily benefit from the same firm strategies (Venkatraman, 1989). Further, scholars have argued that IT alignment does not have an end date but requires continuous change to meet ongoing competitive forces (Henderson, & Venkatraman, 1993). Supplementary findings suggest organic firm structures amid quickly evolving business environments may be beneficial (McAdam et al., 2019). DevOps shares many of the same attributes from both a structural and relational viewpoint. For example, DevOps emphasizes continuous change in technology and internal department collaboration. Thus, in context of the contingency theory, DevOps needs further investigation if business-IT alignment influences its practice (Hart, 2017).

Despite early efforts to determine a central definition of DevOps, further empirical and theoretical efforts are necessary to measure the value that DevOps adds, if any, to organizations. Despite the existing body of DevOps literature not having a significant emphasis on strongly researched IT frameworks and theories (Erich et al., 2017; Jabbari et al., 2016), IT alignment positively associates with its practices thus far (Hart, 2017; Iden et al., 2012). Moreover, few DevOps studies that include theory use quantitative methods to evaluate this value (Jabbari et al., 2016). As a result, this study seeks to further existing empirical work by extending and quantitatively testing a DevOps IT alignment model.

The DevOps IT Alignment Model

One of the first quasi-experimental investigations into the value of DevOps is the DevOps IT alignment model. Findings indicated that while continuous integration of software and cross-discipline collaboration correlate with IT subunit alignment, knowledge sharing does not have a significant influence on departmental alignment (Hart, 2017). This paper repeats two categories of IT alignment, the structural and relational dimensions. It also extends the experimental DevOps IT Alignment model through the inclusion of the more ubiquitously studied latent construct of business-IT alignment in contrast to subunit alignment between development and operations alone. In other words, it posits that when subunits like software development and IT operations are in alignment, there is a positive impact on business-IT alignment.

Strategic IT Alignment

The influence of technology on the competitive forces of a given market mandate careful assessment and strategic planning (Henderson & Sifonis, 1988). Alignment of information systems strategy and firm strategy supports several dimensions of organizational performance in prior literature and it is a predecessor of firm success in various results (Chan et al., 2006; Chan & Reich, 2007; Chen et al., 2008; Kearns & Lederer, 2003; Kearns & Sabherwal, 2007; Luftman & Brier, 1999; Preston & Karahanna, 2009; Wang et al., 2015). Regardless, professionals often agree that no perfect achievement of alignment exists, as all organizations would follow the associated steps if in existence. Instead, alignment is consistently adapting to help support firm competitiveness and performance (Henderson & Sifonis, 1988; Luftman & Kempaiah, 2007). Despite this adaptation, certain pillars of alignment have proven their importance.

Prior IT alignment research uses categories such as "relational" and "structural" to illustrate how business-IT alignment functions within an organization (Luftman & Kempaiah, 2007; Reich & Benbasat, 2000). Narrower types of alignment, such as IT subunit alignment further categorize shared understanding between business units, partnerships, and the capacity to change as relational aspects of alignment. Structural components of alignment associate with processes or architecture, governance, and subunit measurements (Dhaliwal et al., 2011).

To explore the relational and structural dimensions of IT alignment, the authors found that at least two links exist with properties often associated with the practice of DevOps: continuous integration of software (Dhaliwal et al., 2011) and knowledge sharing (Iden et al., 2012). The most cited definition of DevOps that represents the less technical (knowledge sharing through collaboration of people) and more technical (continuous integration of software) dimensions further support both constructs (Leite et al., 2020).

Continuous Integration as a Structural Dimension of IT Alignment

Structural components of IT subunit alignment include measurements, governance, and processes between firm subunits. Architecture in context of subunit alignment associates with factors such as managing the agility of innovative technology, congruent enterprise architecture, and the implementation of efficient standardization across IT subunits. Measurements within IT subunit alignment associate with sharing metrics between disciplines in a way that continuous improvement can occur (Dhaliwal et al., 2011).

Two of the subsequent principles of DevOps, automation of development and operations processes, as well as the measurement of processes, may associate with the structural dimension of alignment (Colomo-Palacios et al., 2018; Humble & Molesky, 2011). Varying architectures, such as cloud and hybrid cloud environments, make automating tools and processes for continual deployment of software to customers even more challenging. DevOps helps bridge the gap between traditional software development methods and more agile and continuous deployment strategies important to addressing new cloud architectures (Metzger et al., 2017; Wettinger et al., 2016). Previous literature associates the standardization of enterprise architecture and the congruency of software development processes with the structural dimension of IT alignment (Dhaliwal et al., 2011). Within DevOps, common themes include automation of deployment and relational software development processes, continuous delivery, and measurement (Bang et al., 2013; Colomo-Palacios et al., 2018; Gupta et al., 2017; Humble & Molesky, 2011; Liu et al., 2014). Although pillars such as measurement and automation in the context of DevOps are not known to directly correlate with factors of alignment at the time of this writing, continuous improvement is a structural concept that both DevOps and IT alignment benefit from in prior research (Bang et al., 2013; Dhaliwal et al., 2011; Liu et al., 2014; Luftman & Kempaiah, 2007). Furthermore, to achieve higher levels of business-IT alignment maturity, continuous improvement between IT and business is necessary (Luftman & Kempaiah, 2007). Therefore, the authors propose that:

> H1. Continuous integration of software code positively influences the level of IT subunit alignment when practicing DevOps.

Knowledge Sharing as a Relational Dimension of IT Alignment

Empirical findings indicate that the achievement of business and IT alignment rest at least partially on both the structural and relational elements of an organization (Luftman & Kempaiah, 2007; Reich & Benbasat, 2000). Relational characteristics of IT subunit alignment include competencies, partnerships, and shared understanding between firm subunits (Dhaliwal et al., 2011). Empirical results show that CEO and CIO participation in relational factors of IT alignment such as business planning are antecedents to IT used to support the competitive advantage of the firm (Luftman & Brier, 1999). Knowledge sharing, planning, and continuous communication are connecting threads between IT alignment and the development and operations subunits (Iden et al., 2012). Furthermore, mapping studies that analyzed volumes of DevOps articles illustrate the significance of integrating IT knowledge areas to achieve its goals (Jabbari et al., 2016).

One of the early relational aspects of IT alignment that ties to IT development and IT operations is shared domain knowledge (Iden et al., 2012). At lower levels of alignment, shared understanding and partnership between organizational units statistically associates with IT subunit alignment. While this DevOps dimension could have certain connections with alignment, it is unexplored in the same context as alignment literature review thus far in this study. For example, it associates with cultural locus of power (Dhaliwal et al., 2011). Although the root ontology of DevOps is still developing, several initial categorizations or pillars of DevOps have emerged. For instance, certain scholars identify knowledge sharing as one of the four primary principles of DevOps, the other three being culture, measurement, and automation (Colomo-Palacios et al., 2018; Humble & Molesky, 2011).

Knowledge sharing is also a relational IT-business alignment factor (Luftman & Kempaiah, 2007). According to certain findings (Luftman & Brier, 1999), knowledge sharing within the strategic alignment process can help generate a competitive advantage for organizations. Knowledge sharing, through both tacit and explicit domain knowledge, positively influences alignment performance. The sharing of discipline-specific information between IT operations, quality assurance, and software development in agile-based environments has emerged as a key driver for effective implementation of DevOps, and particularly when practiced in supporting cloud architectural environments (Colomo-Palacios et al., 2018). Knowledge sharing is therefore a common relational dimension of IT alignment that could be a bridge between prior theory in the IT discipline and new research in the DevOps domain. Thus, this study theorizes that:

> H2. Knowledge sharing between DevOps team members positively influences the level of IT subunit alignment.

Strategic IT Subunit Alignment

Strategic IT subunit alignment is foundational for strategic information systems alignment as it occurs at a lower level in the organization. Researchers have argued that business-IT alignment often assumes that lower levels of the organization are aligned (Dhaliwal et al., 2011), yet findings often indicate that IT subunits have many challenges that may hinder business-IT alignment (Iden et al., 2011, 2012; Zhang et al., 2014). This tapered level of alignment focuses more closely on the internal dimensions of the organization, such as alignment between IT divisions or departments (Dhaliwal et al., 2011). Prior IT subunit alignment models extend certain characteristics of business-IT alignment.

For example, they have modeled parallel relational and structural dimensions of business-IT alignment (Dhaliwal et al., 2011; Luftman & Kempaiah, 2007).

Data triangulated from four studies on the strategic alignment of IT development and IT operations indicated that alignment was challenging and required a "bridge" between the departments (Iden et al., 2012). DevOps could potentially provide such a bridge, due to its ability to foster closer collaboration between these subunits (Bang et al., 2013; Liu et al., 2014). Thus, this study proposes that the practice of DevOps may positively influence the relational and structural dimensions of IT alignment through the practices of knowledge sharing (Iden et al., 2012) and continuous integration of software (Dhaliwal et al., 2011):

> H3. Practicing DevOps between IT subunits within an organization positively influences the level of business-IT alignment.

RESEARCH MEASUREMENTS

This paper addresses two components of IT alignment, the structural and relational dimensions. It also extends the DevOps IT alignment model (Hart, 2017) by testing the more ubiquitously studied latent variable of IT strategic alignment. The investigation of this model requires an exploratory approach. Where possible, the authors of the study use existing validated measures.

Data collection in this study uses a web-based survey with a 7-point Likert scale ranging from "strongly agree" (1) to "strongly disagree" (7). Table 1 identifies the source of each latent variable from the questionnaire.

Table 1. DevOps Latent Variables

Measure	Description	References
Continuous integration	The continuous integration measure identifies whether IT team members integrate new code on a continual basis (Maruping et al., 2009). It is a reflective latent construct.	(Maruping et al., 2009)
Knowledge sharing	Knowledge sharing is a reflective latent construct that measures whether IT team members seek to learn new skills from other team members (van den Hooff & Huysman, 2009; Xiang et al., 2013). The management of knowledge is one of the pillars of DevOps on the basis of principles that include the ability of IT developers and IT operations to understand each other's processes (Iden et al., 2012) and skills (Bang et al., 2013), and to increase software quality (Colomo-Palacios et al., 2018).	(van den Hooff & Huysman, 2009; Xiang et al., 2013)
IT subunit alignment	The IT subunit alignment construct is a reflective latent variable that evaluates whether the IT development subunit and the IT operations subunit strategies, governance, and resources are in alignment (Dhaliwal et al., 2011; Preston & Karahanna, 2009). Prior exploratory findings unveiled four factors that these subunits need to cooperate for alignment to occur including knowledge, communication, planning, and partnership (Iden et al., 2012). DevOps research indicates that knowledge sharing is a primary driver of continuous integration between software development and the operational deployment of applications (Colomo-Palacios et al., 2018).	(Dhaliwal et al., 2011; Preston & Karahanna, 2009)

Measure	Description	References
Business-IT alignment	Strategic IT alignment measures whether the information system strategy is congruent with the organization's strategy (Preston & Karahanna, 2009). It is a reflective latent variable. This study explores whether the subunit alignment of IT development and IT operations through the practice of DevOps can solve certain mis-alignments through the relational and structural dimensions of IT alignment. The theory is that stronger lower level organizational alignment could lead to improved business-IT alignment.	(Preston & Karahanna, 2009)

The DevOps IT alignment model includes control variables that influence alignment in prior research. Table 2 outlines the control variables. These include subunit size, the number of years of employment, and the amount of DevOps experience (Dhaliwal et al., 2011; Preston & Karahanna, 2009).

Table 2. Control Variables

Measure	Description	Reference
Subunit size	The size of an organization has the potential to affect how employees view alignment. Prior findings did not find overall firm size influenced the results but was an important control to validate (Dhaliwal et al., 2011). While this study collected data from similar sized organizations, the possibility for IT outsourcing could significantly variate the size of the IT subunits. Therefore, the study controlled for subunit size on strategic IT alignment. The control variable identifies the number of employees within each subunit.	(Dhaliwal et al., 2011)
Company tenure	Length of tenure is a subsequent control variable that prior information systems alignment research recommends assessing (Preston & Karahanna, 2009). The tenure control variable determines the length of employee service at the organization.	(Preston & Karahanna, 2009)
DevOps Experience	DevOps teams are still novel in many organizations and experience in such a culture could influence the levels of strategic alignment of IT with the firm (Preston & Karahanna, 2009). The experience control variable measures the length of time an individual has spent in a DevOps functional role.	(Preston & Karahanna, 2009)

The authors developed the DevOps instrument using existing survey questions, which have prior reliability and validity. The questionnaire is included in Table 3. Confirmatory factor analysis indicated the knowledge sharing measures used in this paper have factor loadings above 0.70 (Xiang et al., 2013). Factor loadings are also above the 0.70 threshold for the IT alignment items in previous results (Preston & Karahanna, 2009). Cronbach alpha scores above the cutoff point of 0.70 is acceptable in factorial loadings within certain exploratory research (Hair et al., 2009). Furthermore, average variance extracted (AVE) scores for the knowledge sharing (Xiang et al., 2013) and IT alignment (Preston & Karahanna, 2009) measures are above 0.50, indicating adequate convergent validity (Hair et al., 2009). Prior reliability analysis of the continuous integration measures shows an *F*-statistic significant at the level of $p < 0.001$ (Maruping et al., 2009). Thus, the authors chose to adapt each of the prior measures.

Table 3. Instrument

Measure	Question	Reference
Continuous integration (CI)	Members of this team integrate newly coded units of software with existing code.	(Maruping et al., 2009)
Continuous integration (CI)	We combine new code with existing code on a continual basis.	(Maruping et al., 2009)
Continuous integration (CI)	Our team does not take time to combine various units of code as they are developed.	(Maruping et al., 2009)
Knowledge sharing (KS)	I like to be kept fully informed of what my colleagues know.	(van den Hooff & Huysman, 2009; Xiang et al., 2013)
Knowledge sharing (KS)	I ask my colleagues about their skills when I want to learn particular skills.	(van den Hooff & Huysman, 2009; Xiang et al., 2013)
Knowledge sharing (KS)	When a colleague is good at something, I ask him/her to teach me.	(van den Hooff & Huysman, 2009; Xiang et al., 2013)
IT subunit alignment (ITSA)	The scope of the IT development group is tightly linked with that of the IT operations group.	(Dhaliwal et al., 2011; Preston & Karahanna, 2009)
IT subunit alignment (ITSA)	The governance of the IT development group is in harmony with that of the IT operations group.	(Dhaliwal et al., 2011; Preston & Karahanna, 2009)
IT subunit alignment (ITSA)	The resources of the IT development group are aligned with those of the IT operations group.	(Dhaliwal et al., 2011; Preston & Karahanna, 2009)
Business-IT alignment (A)	The IS strategy is congruent with the corporate business strategy in your organization.	(Dhaliwal et al., 2011; Preston & Karahanna, 2009)
Business-IT alignment (A)	Decisions in IS planning are tightly linked to the organization's strategic plan.	(Dhaliwal et al., 2011; Preston & Karahanna, 2009)
Business-IT alignment (A)	Our business strategy and IS strategy are closely aligned.	(Dhaliwal et al., 2011; Preston & Karahanna, 2009)

RESEARCH DESIGN

The researchers use structural equation modeling (SEM) to explore whether DevOps factors have any association with more predominant theoretical anchors in information systems and IT literature, namely, business-IT alignment. SEM has several advantages as a second-generation statistical method. Evaluating multiple items through one latent variable is a principal benefit of using SEM, which is important to testing the exploratory DevOps IT alignment model. A subsequent benefit of structural equation modeling is the evaluation of measurement items as well as theory concurrently (Gefen et al., 2011; Hair, Sarstedt, et al., 2014; Lowry & Gaskin, 2014). SEM can test propositions in a theory via multiple paths (Lowry & Gaskin, 2014). For these reasons, the study employs SEM.

Prior to the model specification, several steps occur within the research methodology beginning with sampling and data collection.

Sample

The authors use the Westland (2010) a-priori sampling size formula, detailed in Figure 1 in this paper, which identifies minimum and recommended sample sizes based upon the number of observed and latent variables, probability level, effect size, and the degree of statistical power (Westland, 2010). The Soper (2020) web-based application can verify these results using the latter formula. At a probability level of 0.01, statistical power of 0.80, and large effect size the minimum sample size for this study is 45 while the recommended sample size is 200.

$$n = \max(n_1, n_2)$$

$$A = 1 - \rho^2$$

$$B = \rho \arcsin\left(\frac{\rho}{2}\right)$$

$$C = \rho \arcsin(\rho)$$

$$D = \frac{A}{\sqrt{3 - A}}$$

$$H = \left(\frac{\delta}{z_{1-\frac{\alpha}{2}} - z_{1-\beta}}\right)^2$$

$$n_1 = \left\lceil 50\left(\frac{j}{k}\right)^2 - 450\left(\frac{j}{k}\right) + 1100 \right\rceil$$

$$n_2 = \left\lceil \frac{1}{2H}\left(A\left(\frac{\pi}{6} - B + D\right) + H\sqrt{\left(A\left(\frac{\pi}{6} - B + D\right) + H\right)^2 + 4AH\left(\frac{\pi}{6} + \sqrt{A} + 2B - C - 2D\right)} \right) \right\rceil$$

Figure 1. A-priori Sample Size Formula for SEM (Westland, 2010)

Response Rate and Potential Bias

Participation in this study was both anonymous and voluntary. Instrument protocol outlined guidance for prospective survey participants to assist with the quality and rate of response. It also asked for consent for use in this study prior to administration. To address the possibility of variations in the responses to the survey from a single respondent in a firm, administration protocol recommends multiple respondents per organization. Prior findings suggest same leader participant responses can introduce a degree of unreliability into the results of survey data (Balloun et al., 2011; Pinsonneault & Kraemer, 1993). Thus, the authors of the study sent emails to top IT executives requesting multiple respondent participation in the electronic survey if their organization practices DevOps. Requests went to companies headquartered in the United States on the current Forbes Global 2000 and the Fortune 500. There were two additional follow up email requests after week 3 and week 4 of the initial request to survey DevOps personnel. A total of 57 firms agreed to participate in the study.

According to survey administrators, 206 qualified subjects completed the survey resulting in a response rate of 24%. While not optimal, the response rates of samples in larger organizations can be lower due to several factors. This study did not incentivize, which can negatively affect response rate due to the time investment it takes to complete a survey. Prior studies sending surveys to IT executives at Fortune 500 companies indicate that response rates below 10% is not out of the ordinary (St. John et al., 2014). Similarly, IT alignment research has experienced lower response rates within this range due to insufficient personnel time and the large number of surveys that organizations receive each year inquiring for participation (Preston & Karahanna, 2009).

The authors used the descriptive missing data analysis in the IBM SPSS statistical software platform to identify if the data set included any absent records. Missing data analysis resulted in the removal of two records. This reduced the total number of responses to 204 subjects.

The authors analyzed both nonresponse bias and common method bias to identify any potential issues with the instrumentation used during data collection. First, wave analysis determined if there were differences in early and late survey responders (Armstrong & Overton, 1977; Rogelberg & Stanton, 2007). Time of completion was the divisor between the two sub-groups. The authors compared the means of each sub-group using the independent-samples t-test. Results were based upon the formative modeled measures of the size of the organizational subunit, the number of years of employment of each subject, and the degree of DevOps experience.

Psychological separation exists within each section of the questionnaire between the exogenous and endogenous items to help with the possibility of common method bias during completion of the questionnaire (Podsakoff et al., 2003). A subsequent dimension of common-method bias that is important to review associates with predictors having a level of variance with other measures within structural equation modeling. Full collinearity assesses this issue using variance inflation factors (VIFs) between the latent variables in the structural model (Kock & Lynn, 2012). Particularly when formative constructs exist within partial least squares structural equation modeling, the inner path model necessitates investigation into collinearity disparities (Hair, Sarstedt, Hopkins, & Kuppelwieser, 2014). VIFs over the threshold of 3.3 require further assessment to determine if the variance is influencing the results (Kock & Lynn, 2012). Table 4 highlights the results of this analysis. Length of employee tenure (TEN), organizational size (SIZE), and the amount of DevOps experience (EXP) are each below the recommended VIF threshold.

Table 4. Full Collinearity VIFs

CI	KS	ITSA	A	*SIZE	*TEN	*EXP
1.258	1.106	1.458	1.525	1.058	1.193	1.217

*Asterisks identify formative control variables

Initial Data Analysis

Unless otherwise noted, the authors use current versions of IBM's SPSS Statistics software, SmartPLS, and ScriptWarp System's WarpPLS to perform the data analysis in this study. Prior to pre-processing of the data, the authors reverse coded one continuous integration item, item 3, as the scale was purposely opposite of the other indicators.

Indicators for the latent variables had some degree of skewness and kurtosis. Scholars have noted that measures outside the skewness range of -2.0 to 2.0 could indicate a level of non-normality exists that needs attention (Fabrigar et al., 1999; Miles & Shevlin, 2001). Kurtosis thresholds for critical values are between -2.58 and 2.58 according to certain statistical experts (Hair et al., 2009). The authors used SPSS to calculate both skewness and kurtosis for each measure. The largest values were -1.19 for skewness on the first continuous integration measure and 1.62 for kurtosis on the second item for knowledge sharing. Thus, results did not indicate any measures outside the latter acceptable ranges (Fabrigar et al., 1999; Hair et al., 2009; Miles & Shevlin, 2001).

Authors of the study also reviewed the Shapiro and Wilk test of normality for each item (Wilk & Shapiro, 1965). All indicators were statistically significant at $p < 0.001$ (Hair et al., 2009). Researchers carefully reviewed Q-Q plots for each indicator. Within SPSS, data points are linear in contrast to non-linear across the diagonal line in each graph. Q-Q plots show the items evenly distribute between each of the latter items with no measures falling significantly beyond the line (Hair et al., 2009).

The authors used several examinations to explore assumptions essential to multivariate statistics and structural equation modeling (Hair et al., 2009). This included Mahalanobis distance scores using the cumulative probability application in SPSS. Degrees of freedom were $p < .001$. Mahalanobis distance results showed five cases statistically significant at $p < .001$, which were subsequently removed. This reduced the data set to 199 responses.

Levene's test of equality of error variances is not significant for any of the study variables, $p > 0.05$, which addresses the assumption of homogeneity of variance (Hair et al., 2009; Olkin, 1960). Evaluation of the relationship between each variable using bivariate scatterplots does not suggest homogeneity of the measures that are problematic. Review of Bartlett's test for sphericity to analyze the possibility of multicollinearity produced results of $p < 0.05$ on each of the correlations. The latter results are satisfactory (Bartlett & Fowler, 1937). Furthermore, the authors reviewed the variance inflation factors along with the standard error of the regression coefficients, neither of which have any significant issues. Variance was less than the recommended tolerance level of 3.0 on each of the formative indicators in this study (Hair et al., 2009).

Table 5. Factor Loadings of Indicators

Indicators	**Factor Loadings**			
	1	*2*	*3*	*4*
CI1	0.876			
CI2	0.936			
CI3	0.794			
KS1		0.729		
KS2		0.889		
KS3		0.791		
ITSA1			0.904	
ITSA2			0.965	
ITSA3			0.887	
A1				0.972
A2				0.962
A3				0.896

Oblique rotation. Loadings < 0.3 suppressed.

The authors evaluated factor analysis loadings and cross-loadings of the items from the questionnaire before structural equation modeling to determine if any internal inconsistencies existed prior to the model specification. Analysis includes oblique rotation of the factors. For exploratory research, loadings above the 0.70 threshold are acceptable (Hair et al., 2009). Table 5 delineates the factor loadings. Each of the items loaded on the theorized factor at thresholds greater than 0.7. Cross-loadings of the items did not indicate correlations with unanticipated factors above 0.3.

The latter analysis concludes addressing the multivariate assumptions. Due to missing data and inordinate outliers, the data set of 206 subjects reduced to 199. With these exceptions, the data set met multivariate assumptions at recommended levels within reason. Factor analysis illustrated adequate item loadings on the hypothesized factors. Results also showed levels of reliability and validity consistent with their use in prior studies (Hair et al., 2009; Maruping et al., 2009; Preston & Karahanna, 2009; Xiang et al., 2013). Thus, analysis turns to path modeling to assess the DevOps IT alignment model.

EVALUATION OF THE DEVOPS MODEL

MODEL SPECIFICATION

Hair, Hult, et al. (2014) define three primary steps researchers should take in using partial least squares structural equation modeling (PLS-SEM). This includes model specification, outer model evaluation, and inner model evaluation. Model specification associates with the creation of the initial path model. This development relies upon theory (Hair, Hult, et al., 2014; Hair, Sarstedt, et al., 2014). Model specification should identify two components: 1) the model constructs, and 2) the hypothesized constructs (Henseler et al., 2016).

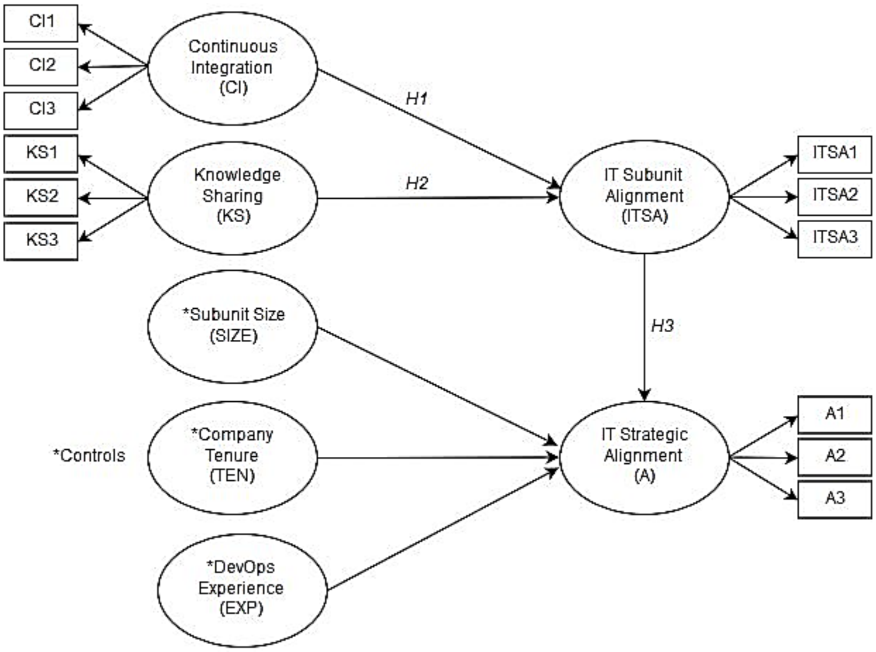

Figure 2. DevOps IT Alignment Model

The path model specified in Figure 2 matriculates from the review of literature and hypothesis development. The measurement instrument as previously indicated in the initial data analysis has shown reliability and validity in prior research. Table 1 outlines the latent variables within the conceptual model and the supporting studies where the items have previous theoretical development. Each of the latent variables model reflectively. The outer model of the exogenous variable continuous integration (CI) includes three items, CI1, CI2, and CI3. These measure the level of continuous integration of programming code that occurs within DevOps teams and whether this has any influence on IT alignment (Hart, 2017; Maruping et al., 2009). Within the conceptual model, the authors theorize that continuous integration could be a structural representation of IT alignment (Colomo-Palacios et al., 2018; Humble & Molesky, 2011). The knowledge sharing (KS) items of KS1, KS2, and KS3 quantify the degree of knowledge that transfers from IT subunits that practice DevOps (van den Hooff & Huysman, 2009; Xiang et al., 2013). In theory, this should support the alignment of organizational departments (Hart, 2017) and a relational dimension of IT alignment (Iden et al., 2012). Therefore, the authors hypothesize that the exogenous variables of continuous integration and knowledge sharing influence the endogenous variable of IT subunit alignment in a DevOps environment.

Three reflective items represent IT subunit alignment (ITSA) in the path model. ITSA1, ITSA2, and ITSA3 determine the degree of strategy harmonization at the IT subunit level (Dhaliwal et al., 2011; Preston & Karahanna, 2009). Prior scholars argue that lower level alignment within the firm should positively influence the overarching business-IT alignment level (Dhaliwal et al., 2011; Iden et al., 2012). Thus, this study proposes that within a DevOps atmosphere a greater degree of IT subunit alignment occurs through the relational dimension of knowledge sharing and the structural dimension of continuous integration of code, which leads to greater information systems and organizational strategic alignment.

To assure the accuracy of alignment measures, prior literature recommends controlling for several organizational factors. Table 2 highlights the control variables within the path model. These variables model formatively. Formative indicators, in contrast to reflective, result in the indicators influencing the construct. In other words, the arrows on the direct paths in the model point from the indicators to the construct.

The authors considered testing the DevOps model using covariance-based structural equation modeling (CB-SEM). However, as prior literature cautions, PLS-SEM is often recommended over CB-SEM when there are formative indicators in the model (Hair, Hult, et al., 2014; Hair, Sarstedt, et al., 2014). More specifically, CB-SEM can result in identification issues when assessing formative indicators (Jarvis et al., 2003). In the case of this study, the control variables model formatively within initial conceptual model analysis using PLS-SEM to determine whether organizational size, the number of company employees, and the experience of DevOps personnel have any impact on the latent variable strategic IT alignment. Thus, the authors continued with PLS-SEM to explore the DevOps IT alignment model. This concludes the model specification, leading to the outer model evaluation of the latent variable items.

Outer Model Evaluation

Outer model evaluation, which is also referred to as the measurement model, explores the reliability and validity of the items in the outer boundary of the path model. Determining how the measurement model is explored has several contingencies given the makeup of the outer model observations (Hair, Sarstedt, et al., 2014). Important to this investigation is whether DevOps has connections to any established domains of literature. This study explores if connections exist with the relational and structural dimensions of IT alignment specifically and whether this has any meaning for the maturing of DevOps as an industry practice. PLS-SEM also allows the authors to assess the entire theoretical model using formatively modeled constructs and latent variables (Gefen et al., 2011; Hair, Sarstedt, et al., 2014; Lowry & Gaskin, 2014).

The outer model or measurement model refers to the relationships of the observed measures and the model constructs (Henseler et al., 2016). Inner model or measurement model assessment relies upon the outer measures and their accuracy in measuring the latent constructs. Thus, it is important to evaluate the measurement model prior to the inner structural model (Hair, Sarstedt, et al., 2014).

Initial outer model evaluation occurs using the PLS regression algorithm (Wold et al., 2001). Analysis included both Cronbach's α as well as composite reliability, which researchers consider a more suitable assessment for internal reliability (Hair, Sarstedt, et al., 2014). Cronbach's alpha coefficients are 0.838 for continuous integration, 0.722 for knowledge sharing, 0.908 for IT subunit alignment, and 0.938 for IT alignment. Each are above the recommended level of reliability at 0.70 or greater (Hair et al., 2009). Fornell and Larcker (1981) suggest adequate composite reliability scores of 0.70 or higher. Composite reliability values are 0.903 for continuous integration, 0.845 for knowledge sharing, 0.942 for IT subunit alignment, and 0.960 for IT alignment. Thus, composite reliability results were also above suggested thresholds for each of the constructs (Fornell & Larcker, 1981; Ringle et al, 2009).

Table 6. Combined Loadings and Cross-Loadings

	CI	KS	ITSA	A	SE	*P*-value
CI1	0.899	0.001	0.141	−0.061	0.060	<0.001
CI2	0.927	0.038	0.000	−0.006	0.059	<0.001
CI3	0.780	−0.046	−0.162	0.077	0.061	<0.001
KS1	0.103	0.684	−0.216	0.097	0.062	<0.001
KS2	−0.003	0.898	0.076	−0.084	0.060	<0.001
KS3	−0.083	0.818	0.097	0.011	0.061	<0.001
ITSA1	0.064	0.008	0.896	−0.043	0.060	<0.001
ITSA2	−0.064	−0.049	0.943	0.012	0.059	<0.001
ITSA3	0.003	0.043	0.918	0.030	0.059	<0.001
A1	−0.003	0.052	−0.101	0.930	0.059	<0.001
A2	−0.035	−0.043	0.017	0.953	0.059	<0.001
A3	0.038	−0.007	0.082	0.946	0.059	<0.001

Standard errors (SEs) and *p*-values are for the loadings.

Table 6 summarizes the combined loadings and cross-loadings for the structural model. Each of the indicators model reflectively, which include continuous integration (CI), knowledge sharing (KS), IT subunit alignment (ITSA), and strategic IT alignment (A). The objective of the loadings is for reflective indicators to have high combined loadings and low cross-loadings (Kock, 2018). P-values within the table are relative to confirmatory factor analysis validation. Standard error (SE) within the Table 6 results are helpful for researchers to observe differences in the structural model coefficients while using the multi-group method (Kock, 2014). Verification of discriminant validity can include identifying whether the loadings of each item are greater than the values of the cross-loadings of the other indicators in the model. Careful observation of the table illustrates cross-loadings are all less than 0.142. In contrast, combined loadings are all above 0.5.

For convergent validity of the measurement instrument, scholars recommend that loadings in Table 6 be equal to or greater than 0.5. Additionally, the *p*-values parallel to the loadings in the table should be either equal to or less than 0.05, which was true for each of the measures in this path model. Another validity assessment is to evaluate each item's average variance extracted (AVE). Researchers also recommend AVE values above 0.50 (Hair, Sarstedt, et al., 2014; Henseler et al., 2016). The average variance extracted value for continuous integration is 0.758, knowledge sharing is 0.648, IT subunit alignment is 0.845, and IT strategic alignment has an AVE of 0.890. Therefore, the results indicate a healthy degree of validity between each of the items and their measurement intention (Kock, 2018; Kock, 2014). This concludes the outer model evaluation. Following these assessments, the authors appraise the inner path model.

INNER MODEL EVALUATION

The inner model evaluates the hypothesized relationships between the exogenous and endogenous variables within the PLS analysis. Prior to assessing the results of the hypotheses, several steps are important to assuring a valid model fit exists. As scholars indicate (Henseler et al., 2016), there are several limitations to fitting PLS models, and it is therefore important not to rely upon single fit indices. The goodness-of-fit (GoF) of the model is an important contingency of the latter. Tenenhaus et al. (2005) outline GoF and its usage for identifying a level of explanatory power in PLS models. Results from the DevOps IT alignment model illustrate a GoF of 0.437. The outputted values correlate with Cohen's (1988) effect sizes where values greater than or equal to 0.36 have a large effect size. A GoF below 0.1 is too low for acceptable explanatory power (Kock, 2018). Despite this explanatory power, GoF is limited as a goodness of fit measure and new criterion such as mean square error correlation provides additional evaluation criterion (Henseler et al., 2016).

Additionally, comparing the fit indices to both the estimated model as well as the saturated model is common to uncover any variations in fit when all constructs are extemporaneously associated in the model (Henseler et al., 2016). GoF of the saturated model is lower than the estimated model with a value of 0.350. In contrast to the estimated model, the saturated model has a medium effect size but is still above the 0.1 level (Kock, 2018).

Simpson's paradox is another important test to determine if different signs exist between correlated variables and a path coefficient (Kock, 2018; Kock & Gaskins, 2016). Sympson's paradox ratio (SPR) should be equal to or greater than 0.7 for the model to be satisfactory (Kock, 2018). SPR for the estimated model is 1.0 and 0.778 for the saturated model, which are both acceptable.

Henseler et al. (2016) also recommend assessment of approximate model fit using the standardized root mean square residual (SRMR). SRMR calculates differences between empirical correlation matrices and the model-implied, which uses the loadings and weights. Furthermore, the standardized mean absolute residual (SMAR) assesses differences between each of these matrices. Values less than 0.1 for the standardized root mean squared residual and the standardized mean absolute residual are necessary for adequate model fit (Kock, 2018). SRMR = 0.079 and SMAR = 0.056 for the estimated and saturated models, which are less than the value of 0.1. These evaluations conclude the model fit assessment, leading to the inner model analysis.

PLS-SEM scholars recommend that inner model evaluation should minimally include path coefficients (β), effect size (f^2), coefficient determination (R^2), and cross-validated redundancy (Q^2) (Hair, Sarstedt, et al., 2014). Figure 3 outlines the PLS-SEM results. Path coefficients of the relationships exist above the associated paths from the exogenous variables to endogenous variables. *P*-values below 0.001 are identified in the path model by double asterisks adjacent to the path coefficients. Coefficient determinations exist within the latent variables of IT subunit alignment and IT strategic alignment.

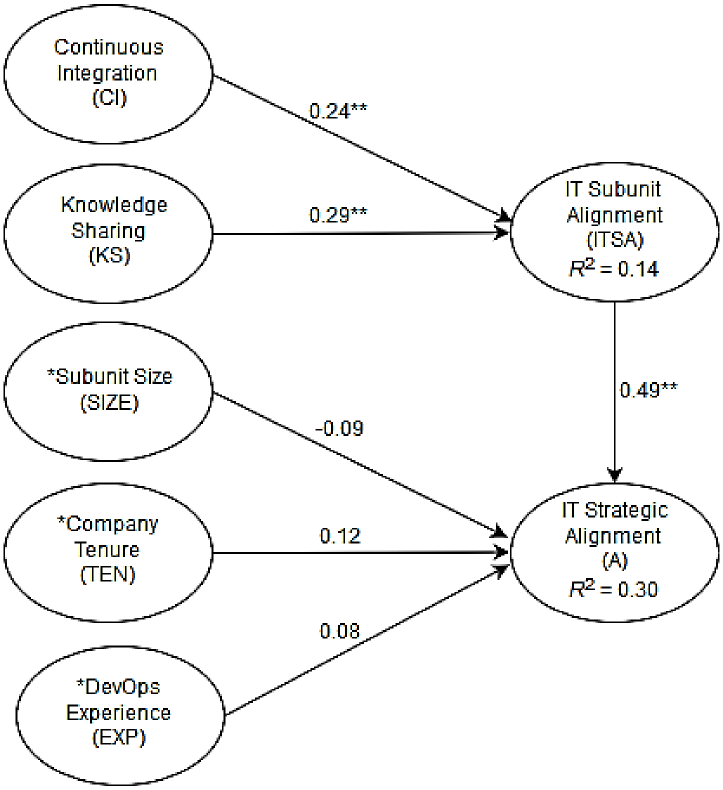

Figure 3. ** $p < 0.001$. * Control variables.

DevOps IT Alignment Model PLS Results

Each of the hypothesized relationships in the path model indicate some degree of statistical significance according to their path coefficients and *p*-values. Path coefficients standardize from -1.0 to 1.0, where a coefficient value that is closer to either has a stronger relationship (Hair, Sarstedt, et al., 2014). Effect size quantifies the impact of the effect and often follow Cohen's (1988) thresholds. Effect size values beyond 0.02 are small, values greater than 0.15 are moderate, and values above 0.35 are strong (Hair et al., 2017; Henseler et al., 2016).

Continuous integration of software in organizations that practice DevOps correlates with IT subunit alignment but with a small effect size ($\beta = 0.241, p < 0.001, f^2 = 0.054$). Knowledge sharing also has a significant association with IT subunit alignment with a small effect size ($\beta = 0.294, p < 0.001, f^2 = 0.083$). Finally, IT subunit alignment significantly relates to the strategic alignment of business and IT ($\beta = 0.493, p < 0.001, f^2 = 0.254$). In this case, the effect size is strong.

Before reviewing measures correlated with the endogenous constructs, the authors assess the control variables and their potential influence on the model. The non-theorized paths occur between IT strategic alignment and the control variables of subunit size, company tenure, and DevOps experience. None of the control variable associations are statistically significant at $p < 0.01$; subunit size ($\beta = -0.088, p = 0.104, f^2 = 0.008$), company tenure ($\beta = 0.116, p = 0.049, f^2 = 0.021$), and DevOps experience ($\beta = 0.082, p = 0.120, f^2 = 0.017$). Additionally, the effect sizes of each of the control variables in the path model have small consequences. Therefore, the control variables do not significantly influence IT strategic alignment. This leads to the final inner model analyses of coefficient determination and cross-validated redundancy.

Cross-validated redundancy parallels a nonparametric investigation into the structural model's ability to predict a removed portion of the data matrix by utilizing estimates (Hair, Sarstedt, et al., 2014). More specifically, it refers to the latent variable block in the PLS-SEM model (Kock, 2014, 2018). Q^2 notates the difference between the data matrix estimates and the normal results without a modified data matrix. Smaller values between the different matrices correlate with larger Q^2 coefficients and subsequently a greater amount of predictive relevance of the structural model (Hair, Sarstedt, et al., 2014). Values of Q^2 greater than zero illustrates an adequate degree of predictive accuracy for the associated endogenous latent variable (Hair, Sarstedt, et al., 2014; Kock, 2014). Cross-validated redundancy results for the endogenous variable IT subunit alignment show adequacy where $Q^2 = 0.139$. The IT strategic alignment construct is also sufficient as $Q^2 = 0.301$, indicating a greater degree of predictive relevancy comparatively. Thus, each of the endogenous latent variables have some level of predictive accuracy. This leads to coefficient of determination.

Coefficient of determination quantifies predictive accuracy of the path model through measurement of the exogenous variable's effects on the endogenous variables. It helps explain the amount of variance of the latent variables. The value of the effect should be between 0 and 1, where 1 associates with complete accuracy (Hair, Sarstedt, et al., 2014). The degree of variance and its acceptable degree of impact depends on the path model. For instance, several exogenous variables can have direct links to one endogenous variable. In the latter case, a larger coefficient of determination would be necessary than an inner model where only one exogenous variable has a direct link to the endogenous variable (Hair et al., 2017; Ringle et al, 2009). Considering adequate R^2 depends on the number of exogenous variables. Guidance thresholds include 0.19, 0.33, and 0.67 (Ringle et al, 2009) or the levels 0.25, 0.50, and 0.75 for weak, moderate, and substantial (Hair, Sarstedt, et al., 2014). Notable is that this model is small and therefore likely explains a minor percentage of variance comparatively to a potentially larger DevOps IT alignment model. Figure 3 illustrates that $R^2 = 0.14$ for the endogenous variable IT subunit alignment, which is below a weak level of explained variance given traditional recommendations. Subsequently, $R^2 = 0.30$ for the IT strategic alignment construct. This illustrates a weak level of expected accuracy (Hair, Sarstedt, et al., 2014; Ringle et al, 2009).

Coefficient of determination is known to have weaknesses such as spurious increases in R^2 due to inflationary model additions and thus researchers should use alternative analyses (Hair, Sarstedt, et al., 2014; Kock, 2018; Wooldridge, 1991). In the case of the estimated model, several control variables exist that could negatively influence the results. Adjusted R^2 is an alternative measurement that helps regulate negligible model additions by reducing the R^2 value as latent variables increase (Hair, Sarstedt, et al., 2014). Suggested thresholds of adjusted coefficient of determination follow Cohen's (1988) levels where the values must be above 0.02, 0.15, and 0.35 for small, medium, and large effects (Kock, 2018). Accordingly, adjusted $R^2 = 0.128$ for the IT subunit latent construct. This exemplifies a small effect size. Adjusted $R^2 = 0.284$ for the IT strategic alignment endogenous variable. The latter result parallels a medium effect size.

HYPOTHESIS RESULTS

The results of the inner model evaluation indicate the model has an acceptable degree of explanatory power. Like the model size, the amount of variance in the latent variable blocks is small. Although the control variables have no substantial associations with IT alignment, each of the hypothesized relationships in the path model are statistically significant.

Table 7 summarizes the hypothesis results. Continuous integration of software code and knowledge sharing both positively influence the level of IT subunit alignment. While the R^2 value for the endogenous IT subunit alignment variable has no substantial effect, its adjusted R^2 associates with a weak level of explained variance. Subsequently, practicing DevOps between IT subunits positively influences the level of business-IT alignment. Both R^2 and adjusted R^2 values for IT strategic alignment show sufficient levels of predictive accuracy. This concludes the evaluation of the IT DevOps alignment model.

Table 7. Hypothesis Results

Hypothesis	Relationship	Result
*H*1	Continuous integration of software code positively influences the level of IT subunit alignment	Supported
H2	Knowledge sharing positively influences the level of IT subunit alignment	Supported
H3	Practicing DevOps between IT subunits positively influences the level of business-IT alignment	Supported

DISCUSSION

The authors' findings are consistent with several prior studies. First, a reliable definition of DevOps is necessary to progress a key objective of this research, identifying connections with prior theory (Leite et al., 2020; Jabbari et al., 2016). As scholars collectively agree upon and more consistently cite a common definition of DevOps, future work should study the ontology represented by its higher-level concepts. This could lead to valuable mapping studies with relevant theories.

Second, while this study adds to prior findings that indicate the practice of DevOps in large U.S. firms results in stronger sub-unit alignment (Hart, 2017), further studies should validate this result (Erich et al., 2014a). Continuous integration of software has emerged as a positive performance measure of IT operations. Perhaps its classification as a capability, referred to as a centerpiece of DevOps (Smeds et al., 2015), lends itself as a metric that adequately represents IT operations and IT development subunits that are higher performers. This is rather significant given the more technical measures of IT performance such as IT infrastructure are often weaker indicators of business-IT alignment (e.g., Dhaliwal et al., 2011). Possibly less surprisingly but no less significant, knowledge sharing between IT subunits that have ominously different technical skillsets and goals, is an important measure of strategic alignment between business and IT. Similar to Dhaliwal et al. (2011), who tested the relational and structural dimensions of strategic IT alignment between software developers and testers, the findings of this study found that sharing knowledge results in stronger sub-unit alignment. A practicing DevOps environment and the addition of business-IT alignment in the PLS-SEM model are two key differences.

Third, a momentous outcome of this study is that practicing DevOps leads to sub-unit alignment, which subsequently results in the positive strategic alignment of business and IT. While prior research has parallel findings on the positive impact of sub-unit alignment between software developers and testers (Dhaliwal et al., 2011) as well as in organizations where DevOps is practiced (Hart, 2017), prior studies stop short of testing business-IT alignment.

Fourth, results highlight statistically significant connections of DevOps to strategic IT alignment. This further validates the value that DevOps adds to practicing organizations. It also provides new avenues for researchers to explore the underlying theories that increase IT performance and the competitive advantages it supports when business and IT are in alignment.

CONCLUSION

THEORETICAL IMPLICATIONS

This study focuses on several building blocks important to the future exploration of DevOps. Research continues to expand in this discipline, which is encouraging due to the ongoing success of DevOps implementations in organizations (Erich et al., 2017). Qualitative results indicate that companies are positive about their experience in implementing DevOps to compete in markets that require faster software delivery times and higher quality applications (Erich et al., 2014a, 2017). However, as previous literature recommends (Erich et al., 2014a, 2014b, 2017), it is important for empirical work to continue to measure whether DevOps is indeed achieving what it promises through quantitative methods. This study addresses this specific building block. It does so by approaching potential underpinnings of this practice with factors proven in prior investigations that lean heavily upon software development theory.

Using the quantitative PLS-SEM methodology, results confirm that certain Global 2000 and Fortune 500 organizations who practice DevOps experience a greater level of IT strategic alignment at the subunit level. Subunit alignment subsequently has a significant correlation with business-IT alignment in the tested model. The results are a promising breakthrough for future DevOps and IT alignment research. Continuous integration of software and knowledge sharing within companies that practice DevOps help firms better align their business and IT strategy at both the lower and higher levels of their organization. As an indicator of varying aspects of firm performance in prior literature (Chan et al., 2006; Chan & Reich, 2007; Chen et al., 2008; Kearns & Lederer, 2003; Kearns & Sabherwal, 2007; Luftman & Brier, 1999; Preston & Karahanna, 2009; Wang et al., 2015), IT strategic alignment is a beneficial measure of the value that organizations receive from nurturing a DevOps culture.

Two categories exist for the relational and structural dimensions of IT alignment, namely, knowledge sharing and continuous integration of software. Both indicate that DevOps has a positive influence on organizations. These results signify that stronger interdepartmental collaboration to integrate software positively benefits knowledge sharing between dissimilar subunits. Additionally, subunit alignment has a significant effect on the alignment of IT strategy and business strategy in large companies, where competition is often fierce. This provides further evidence for the increased value that DevOps provides to organizations through increased continuous integration of software and effective knowledge sharing between dissimilar business units.

As a contribution to future studies, results indicate theories such as the contingency theory mandate further investigation in the information systems domain. As it posits, there is not a unique management structure for all businesses, and it must adapt over time. Rather than a one size fits all management strategy, adaptation becomes more important as the complexity of the business environment increases (McAdam et al., 2019). Similarly, DevOps lacks an anchor in theory. Central to its formation is adaptability throughout the software development lifecycle (Erich et al., 2017). Therefore, future work in management information systems should continue investigations into the contingency theory given the positive influence of DevOps on business-IT alignment (McAdam et al., 2019).

MANAGERIAL IMPLICATIONS

Considering the limitations and threats to validity, there are several valuable lessons outlined in the results of this study. Large organizations play a significant role in the contribution to research. Volunteering time is a challenge. Despite mediocre response rates, the strong teamwork between industry and academia in this study produced non-incentivized results. This has the potential to reduce response bias. While successful, this study illustrates the challenges of sampling and response rates. Academia and industry need to continue to explore creative programs that allow for collaborative and accurate research that adequately benefits all stakeholders. Although certain academic research programs offset corporate R&D, additional disciplines should follow successful models.

This study also supports managerial decisions to implement DevOps teams, particularly in rapidly changing business environments. DevOps adds more value than shortening the software development lifecycle. Although additional research is necessary to address variations in the performance and competitive advantage implications of strategic IT alignment, at the minimum large firms benefitted from a greater level of alignment at the lower and higher hierarchical levels of their organizational structure.

Corporations headquartered in the U.S. showed significant level of inter-departmental alignment when team members shared knowledge between each other and continually integrated their business applications. The ability of DevOps to shorten product lifecycles is undoubtedly appealing to managers. However, several underlying shifts are necessary to make this transition successful. At least two revealed in these results are structural and relational, also identified as technological and cultural. Both the technologies of the organization and its cultural behaviors have influence on the relationship of departments and the overall strategic alignment of business and IT. Therefore, organizations should closely monitor decisions that fail to first assess relational and structural impacts prior to and following DevOps implementation. Differing behaviors effect the technological tools that will be successful. For instance, a culture where knowledge sharing is not progressing continuously between sub-units requires technologies with the supporting features for effective cross-functional comprehension and human resource policies that reward personnel fostering these positive changes.

LIMITATIONS AND FUTURE WORK

Several limitations and threats to validity exist in this study. The sample is not generalizable to broader populations. Future work could repeat or extend this research within companies of varying sizes and with international headquarters. While this study makes an incremental contribution by illustrating an association of DevOps factors with IT strategic alignment, few studies exist that engage in theory development within this practice. Therefore, the authors recommend investigation into the foundations of DevOps with the contingency theory and other related practices such as Quality Management theory (McAdam et al., 2019).

A considerable limitation of this study is the small size of the experimental model. This intentional objective saved valuable time for participants of the research, who completed the questionnaire with no incentives. Supplementary DevOps factors are necessary beyond continuous integration and knowledge sharing to test the possible effects of DevOps on strategic alignment at various levels of the firm. Existing literature (Gupta et al., 2017) has begun this process and this exploratory work should continue.

An important threat within this study pertains to the validity of the constructs and their intended measurement. Latent variables create a level of abstraction from the measuring indicators. As Lowry and Gaskin (2014) outline, indirect measures such as a satisfaction item from a survey that act as a substitute because satisfaction cannot be measured directly are naturally prone to measurement error when used to collect data. First generation statistical methods may test the theoretical relationships of the constructs using separate methods. This departure of theory and evaluation introduce the possibility of inaccurate measurements and ensuing analysis of the results. Two-step approaches have been known to differentiate validity when measures unite with other items (Lowry & Gaskin, 2014).

Structural equation modeling is beneficial as theory development and testing becomes more multifaceted. This is because it addresses both convergent and discriminant validity between the measures concurrently, which associate theoretically in the structural model. The authors addressed more detailed quantitative validity testing in the initial data analysis section of this study. For example, average variance extracted results meet recommended levels of discriminant validity and cross loadings fail to threaten convergent validity at significate correlational levels. Beyond discriminant and convergent validity, there are also several tests performed during hypothesis testing to validate the hypothesized model including coefficient determination, cross-validated redundancy, effect size, and analysis of the path coefficients (Hair, Sarstedt, et al., 2014).

While the methodology in this study attempts to address threats to internal validity, several limitations exist. For instance, the endogenous latent variable of IT alignment has control variables that include the size of the relational IT departments, length of service, and the amount of DevOps experience. The controls did not have significant effects on the model. However, adequate research is not present to address causality or generalizability in this study. A richer theoretical foundation needs to exist for more in-depth analysis of causality, yet as this study outlines the DevOps domain is still in an infant state within scholarly literature. To address generality of the findings, the model did extend from related literature to Fortune 500 and Global 2000 organizations. The sample is specific to large U.S. firms, however, where organizations may be more likely to have practiced DevOps for a longer period. Thus, further work is necessary to address the latter external and internal validity concerns.

In conclusion, the DevOps IT alignment model needs maturing. For instance, there are opportunities to bridge the gap between the cross-discipline knowledge sharing culture of DevOps and the automation of the IT infrastructure (Wettinger et al., 2015). Likewise, Shahin et al. (2018) affirm that continuous delivery of code necessitates more than automation and tool improvements. Scholars indicate that the most significant challenge for implementing continuous delivery and deployment is the rigidity of corporate structures (Shahin et al., 2018). Thus, the authors recommend a more comprehensive DevOps IT alignment model that explores supporting IT infrastructure as well as organizational structure and their influence on continuous delivery and deployment.

REFERENCES

Airaj, M. (2016). Enable cloud DevOps approach for industry and higher education. *Concurrency and Computation: Practice and Experience, 29*(5), e3937. https://doi.org/10.1002/cpe.3937

Armstrong, J. S., & Overton, T. S. (1977). Estimating nonresponse bias in mail surveys. *Journal of Marketing Research*, *14*(3), 396–402. https://doi.org/10.2307/3150783

Balloun, J., Barrett, H., & Weinstein, A. (2011). One is not enough: The need for multiple respondents in survey research of organizations. *Journal of Modern Applied Statistical Methods*, *10*(1), 287-299. https://doi.org/10.22237/jmasm/1304223900

Bang, S. K., Chung, S., Choh, Y., & Dupuis, M. (2013). A grounded theory analysis of modern web applications: Knowledge, skills, and abilities for DevOps. *Proceedings of the 2nd Annual Conference on Research in Information Technology*, 61–62. https://doi.org/10.1145/2512209.2512229

Bartlett, M. S., & Fowler, R. H. (1937). Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, *160*(901), 268–282. https://doi.org/10.1098/rspa.1937.0109

Chan, Y. E., & Huff, S. L. (1993). Strategic information systems alignment. *Business Quarterly*, *58*(1), 51–55.

Chan, Y. E., & Reich, B. H. (2007). IT alignment: What have we learned? *Journal of Information Technology*, *22*(4), 297–315. https://doi.org/10.1057/palgrave.jit.2000109

Chan, Y. E., Sabherwal, R., & Thatcher, J. B. (2006). Antecedents and outcomes of strategic IS alignment: An empirical investigation. *IEEE Transactions on Engineering Management*, *51*(3), 27–47. https://doi.org/10.1109/TEM.2005.861804

Chen, R. S., Sun, C. M., Helms, M. M., & Jih, W. J. (2008). Aligning information technology and business strategy with a dynamic capabilities perspective: A longitudinal study of a Taiwanese semiconductor company. *International Journal of Information Management*, *28*(5), 366–378. https://doi.org/10.1016/j.ijinfomgt.2008.01.015

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Lawrence Erlbaum Associates.

Colomo-Palacios, R., Fernandes, E., Soto-Acosta, P., & Larrucea, X. (2018). A case analysis of enabling continuous software deployment through knowledge management. *International Journal of Information Management*, *40*, 186–189. https://doi.org/10.1016/j.ijinfomgt.2017.11.005

Dhaliwal, J., Onita, C. G., Poston, R., & Zhang, X. P. (2011). Alignment within the software development unit: Assessing structural and relational dimensions between developers and testers. *The Journal of Strategic Information Systems*, *20*(4), 323–342. https://doi.org/10.1016/j.jsis.2011.03.001

Drazin, R., & Van de Ven, A. H. (1985). Alternative forms of fit in contingency theory. *Administrative Science Quarterly*, *30*(4), 514–539. https://doi.org/10.2307/2392695

Erich, F., Amrit, C., & Daneva, M. (2014a). A mapping study on cooperation between information system development and operations. In A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch, & M. Raatikainen (Eds.), *Product-focused software process improvement* (pp. 277–280). Springer International Publishing. https://doi.org/10.1007/978-3-319-13835-0_21

Erich, F., Amrit, C., & Daneva, M. (2014b). Cooperation between information system development and operations: A literature review. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 1). New York, NY, USA: ACM. https://doi.org/10.1145/2652524.2652598

Erich, F., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*, *29*(6), e1885. https://doi.org/10.1002/smr.1885

Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., & Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods*, *4*(3), 272–299. https://doi.org/10.1037/1082-989X.4.3.272

Fokaefs, M., Barna, C., & Litoiu, M. (2017). From DevOps to BizOps: Economic sustainability for scalable cloud applications. *ACM Transactions on Autonomous and Adaptive Systems*, *12*(4), 25:1–25:29. https://doi.org/10.1145/3139290

Fornell, C., & Larcker, D. F. (1981). Structural equation models with unobservable variables and measurement error: Algebra and statistics. *Journal of Marketing Research*, *18*(3), 382–388. https://doi.org/10.2307/3150980

Gefen, D., Rigdon, E. E., & Straub, D. (2011). Editor's comments: An update and extension to SEM guidelines for administrative and social science research. *MIS Quarterly*, *35*(2), iii–xiv. https://doi.org/10.2307/23044042

Gupta, V., Kapur, P. K., & Kumar, D. (2017). Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling. *Information and Software Technology*, *92*, 75–91. https://doi.org/10.1016/j.infsof.2017.07.010

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2009). *Multivariate data analysis* (7th ed.). Prentice Hall.

Hair, J. F., Hollingsworth, C. L., Randolph, A. B., & Chong, A. Y. L. (2017). An updated and expanded assessment of PLS-SEM in information systems research. *Industrial Management & Data Systems*, *117*(3), 442–458. https://doi.org/10.1108/IMDS-04-2016-0130

Hair, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2014). A primer on partial least squares structural equation modeling. Sage.

Hair, J. F., Sarstedt, M., Hopkins, L., & Kuppelwieser, V. (2014). Partial least squares structural equation modeling (PLS-SEM): An emerging tool in business research. *European Business Review*, *26*(2), 106-121. https://doi.org/10.1108/EBR-10-2013-0128

Hart, M. (2017). *IT software development and IT operations strategic alignment: An agile DevOps model* (UMI No. 10260064) [Doctoral dissertation]. ProQuest dissertations and Theses database.

Henderson, J. C., & Sifonis, J. G. (1988). The value of strategic IS planning: Understanding consistency, validity, and IS markets. *MIS Quarterly*, *12*(2), 187–200. https://doi.org/10.2307/248843

Henderson, J. C. & Venkatraman, H. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, *32*(1), 472–484. https://doi.org/10.1147/sj.382.0472

Henseler, J., Hubona, G. S., & Ray, P. A. (2016). Using PLS path modeling in new technology research: Updated guidelines. *Industrial Management and Data Systems*, *116*, 2–20. https://doi.org/10.1108/IMDS-09-2015-0382

Humble, J., & Molesky, J. (2011). Why enterprises must adopt DevOps to enable continuous delivery. *Cutter IT Journal*, *24*(8), 6.

Iden, J., Tessem, B., & Päivärinta, T. (2011). Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts. *Information & Software Technology*, *53*, 394–406.

Iden, J., Tessem, B., & Paivarinta, T. (2012). IS development & IT operations alignment in system development projects: A multi-method research. *International Journal of Business Information Systems*, *11*(3), 343–359. https://doi.org/10.1504/IJBIS.2012.049570

Jabbari, R., Ali, N. B., Petersen, K., & Tanveer, B. (2016). What is DevOps?: A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016* (pp. 1–11). Edinburgh, Scotland UK. https://doi.org/10.1145/2962695.2962707

Jarvis, C. B., MacKenzie, S. B., & Podsakoff, P. M. (2003). A critical review of construct indicators and measurement model misspecification in marketing and consumer research. *Journal of Consumer Research*, *30*(2), 199–218. https://doi.org/10.1086/376806

Kearns, G. S., & Lederer, A. L. (2003). A resource-based view of strategic IT alignment: How knowledge sharing creates competitive advantage. *Decision Sciences*, *34*(1), 1–29. https://doi.org/10.1111/1540-5915.02289

Kearns, G. S., & Sabherwal, R. (2007). Strategic alignment between business and information technology: A knowledge-based view of behaviors, outcome, and consequences. *Journal of Management Information Systems*, *23*(3), 129–162. https://doi.org/10.2753/MIS0742-1222230306

Kock, N. (2014). Advanced mediating effects tests, multi-group analyses, and measurement model assessments in PLS-based SEM. *International Journal of e-Collaboration*, *10*(1), 1–13. https://doi.org/10.4018/ijec.2014010101

Kock, N. (2018). *WarpPLS User Manual: Version 6.0*. ScriptWarp Systems.

Kock, N., & Gaskins, L. (2016). Simpson's paradox, moderation and the emergence of quadratic relationships in path models: An information systems illustration. *International Journal of Applied Nonlinear Science*, *2*(3), 200–234. https://doi.org/10.1504/IJANS.2016.077025

Kock, N., & Lynn, G. S. (2012). Lateral collinearity and misleading results in variance-based SEM: An illustration and recommendations. *Journal of the Association for Information Systems*, *13*(7), 546-580. https://doi.org/10.17705/1jais.00302

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2020). A survey of DevOps concepts and challenges. *ACM Computing Surveys, 52*(6), Article 127. https://doi.org/10.1145/3359981

Li, Q., Wang, C., Wu, J., Li, J., & Wang, Z.-Y. (2011). Towards the business–information technology alignment in cloud computing environment: An approach based on collaboration points and agents. *International Journal of Computer Integrated Manufacturing*, *24*(11), 1038–1057. https://doi.org/10.1080/0951192X.2011.592994

Liu, Y., Li, C., & Liu, W. (2014). Integrated solution for timely delivery of customer change requests: A case study of using DevOps approach. *International Journal of U- & E-Service, Science & Technology*, *7*(2), 41-50. https://doi.org/10.14257/ijunesst.2014.7.2.04

Lowry, P. B., & Gaskin, J. E. (2014). Partial Least Squares (PLS) Structural Equation Modeling (SEM) for building and testing behavioral causal theory: When to choose it and how to use it. *IEEE Transactions on Professional Communication*, *57*, 123-146. https://doi.org/10.1109/TPC.2014.2312452

Luftman, J., & Brier, T. (1999). Achieving and sustaining business-IT alignment. *California Management Review*, *42*(1), 109–122. https://doi.org/10.2307/41166021

Luftman, J., & Kempaiah, R. M. (2007). An update on business-IT alignment: "A line" has been drawn. *MIS Quarterly Executive*, *6*(3), 165-177.

Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, *20*(3), 377-399. https://doi.org/10.1287/isre.1090.0238

McAdam, R., Miller, K., & McSorley, C. (2019). Towards a contingency theory perspective of quality management in enabling strategic alignment. *International Journal of Production Economics*, *207*, 195–209. https://doi.org/10.1016/j.ijpe.2016.07.003

Metzger, S., Durden, D., Sturtevant, C., Luo, H., Pingintha-Durden, N., Sachs, T., Serafimovich, A., Hartmann, J., Li, J., Xu, K., & Desai, A. R. (2017). Eddy4R 0.2.0: A DevOps model for community-extensible processing and analysis of eddy-covariance data based on R, Git, Docker, and HDF5. *Geoscientific Model Development*, *10*(9), 3189–3206. https://doi.org/10.5194/gmd-10-3189-2017

Miles, J., & Shevlin, M. (2001). *Applying regression & correlation: A guide for students and researchers*. Sage Publications.

Olkin, I. (1960). *Contributions to probability and statistics: Essays in honor of Harold Hotelling*. Stanford University Press.

Pinsonneault, A., & Kraemer, K. L. (1993). Survey research methodology in management information systems: An assessment. *Journal of Management Information Systems*, *10*(2), 75-105. https://doi.org/10.1080/07421222.1993.11518001

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology*, *88*(5), 879–903. https://doi.org/10.1037/0021-9010.88.5.879

Preston, D. S., & Karahanna, E. (2009). Antecedents of IS strategic alignment: A nomological network. *Information Systems Research*, *20*, 159–179. https://doi.org/10.1287/isre.1070.0159

Reich, B. H., & Benbasat, I. (2000). Factors that influence the social dimension of alignment between business and information technology objectives. *MIS Quarterly*, *24*(1), 81–113. https://doi.org/10.2307/3250980

Ringle, C. M., Sinkovics, R. R., & Henseler, J. (2009). The use of partial least squares path modeling in international marketing. In R. R. Sinkovics & P. N. Ghauri (Eds.), *New challenges to international marketing* (Vol. 20, pp. 277–319). Emerald Group Publishing. https://doi.org/10.1108/S1474-7979(2009)0000020014

Roche, J. (2013). Adopting DevOps practices in quality assurance. *Communications of the ACM*, *56*(11), 38-43. https://doi.org/10.1145/2524713.2524721

Rogelberg, S. G., & Stanton, J. M. (2007). Introduction: Understanding and dealing with organizational survey nonresponse. *Organizational Research Methods*, *10*(2), 195–209. https://doi.org/10.1177/1094428106294693

Shahin, M., Zahedi, M., Babar, M. A., & Zhu, L. (2018). An empirical study of architecting for continuous delivery and deployment. *Empirical Software Engineering*, *24*, 1-48. https://doi.org/10.1007/s10664-018-9651-4

Smeds, J., Nybom, K., & Porres, I. (2015). DevOps: A definition and perceived adoption impediments. In C. Lassenius, T. Dingsøyr, & M. Paasivaara (Eds.), *Agile processes in software engineering and extreme programming* (Vol. 212, pp. 166–177). Springer International Publishing. https://doi.org/10.1007/978-3-319-18612-2_14

Soper, D. S. (2020). A-priori sample size calculator for structural equation models [Software]. https://www.danielsoper.com/statcalc/calculator.aspx?id=89.

St. John, J., Guynes, C. S., & Vedder, R. (2014). The client–vendor offshore relationship: Success factors. *Information Systems Management*, *31*(2), 120-125. https://doi.org/10.1080/10580530.2014.890429

Tenenhaus, M., Vinzi, V. E., Chatelin, Y. M., & Lauro, C. (2005). PLS path modeling. *Computational Statistics & Data Analysis*, *48*(1), 159–205. https://doi.org/10.1016/j.csda.2004.03.005

van den Hooff, B., & Huysman, M. (2009). Managing knowledge sharing: Emergent and engineering approaches. *Information & Management*, *46*(1), 1-8. https://doi.org/10.1016/j.im.2008.09.002

Venkatraman, N. (1989). The concept of fit in strategy research: Toward verbal and statistical correspondence. *Academy of Management Review*, *14*(3), 423–444. https://doi.org/10.5465/amr.1989.4279078

Wang, Y., Chen, Y., & Benitez-Amado, J. (2015). How information technology influences environmental performance: Empirical evidence from China. *International Journal of Information Management*, *35*(2), 160–170. https://doi.org/10.1016/j.ijinfomgt.2014.11.005

Westland, C. J. (2010). Lower bounds on sample size in structural equation modeling. *Electronic Commerce Research and Applications*, *9*(6), 476–487. https://doi.org/10.1016/j.elerap.2010.07.003

Wettinger, J., Andrikopoulos, V., & Leymann, F. (2015). Automated capturing and systematic usage of DevOps knowledge for cloud applications. In *2015 IEEE International Conference on Cloud Engineering* (pp. 60–65). Tempe, AZ: IEEE. https://doi.org/10.1109/IC2E.2015.23

Wettinger, J., Breitenbücher, U., Kopp, O., & Leymann, F. (2016). Streamlining DevOps automation for cloud applications using TOSCA as standardized metamodel. *Future Generation Computer Systems*, *56*, 317–332. https://doi.org/10.1016/j.future.2015.07.017

Wilk, M. B., & Shapiro, S. S. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3–4), 591–611. https://doi.org/10.1093/biomet/52.3-4.591

Wold, S., Trygg, J., Berglund, A., & Antti, H. (2001). Some recent developments in PLS modeling. *Chemometrics and Intelligent Laboratory Systems*, 58(2), 131-150. https://doi.org/10.1016/S0169-7439(01)00156-3

Wooldridge, J. M. (1991). A note on computing r-squared and adjusted r-squared for trending and seasonal data. *Economics Letters*, *36*(1), 49–54. https://doi.org/10.1016/0165-1765(91)90054-O

Xiang, C., Lu, Y., & Gupta, S. (2013). Knowledge sharing in information system development teams: Examining the impact of shared mental model from a social capital theory perspective. *Behaviour & IT*, *32*, 1024–1040. https://doi.org/10.1080/0144929X.2012.745901

Zhang, X., Stafford, T. F., Dhaliwal, J. S., Gillenson, M. L., & Moeller, G. (2014). Sources of conflict between developers and testers in software development. *Information & Management*, *51*(1), 13–26. https://doi.org/10.1016/j.im.2013.09.006

BIOGRAPHIES

Michael Hart is an Assistant Professor of Computer Information Science at Minnesota State University, Mankato. He is a passionate educator that yearns to unite academia and industry where valuable. His research interests include distributed systems, information security, IT management, and parallel systems.

John Burke is an Assistant Professor of Computer Information Science at Minnesota State University, Mankato. He completed his PhD at the University of Illinois at Urbana-Champaign. He currently performs research in the areas of game design and data science.