Master Thesis
for the study program MSc Business Administration

ENHANCING ENTERPRISE AGILITY WITH BIZDEVOPS METHOD FOR BUSINESS-IT ALIGNMENT

Keywords:
BizDevOps, enterprise agility, business-IT alignment, BizDevOps KPI metrics

Author:

MSc candidate:	Nikola Stanković
Study number:	2272245
Study program:	MSc Business Administration
Specialization track:	Digital Business
E-mail:	n.stankovic@student.utwente.nl

Graduation committee:

Lead supervisor:	Prof. Dr. M. E. Iacob (*Faculty of BMS & IEBIS, University of Twente*)
Second supervisor:	Prof. Dr. J. van Hillegersberg (*Faculty of BMS & IEBIS, University of Twente*)
Other supervisors:	PhD candidate O.H. Plant (*Faculty of BMS & IEBIS, University of Twente*)
	Dr. A.I. Aldea (*Faculty of BMS & IEBIS, University of Twente*)

Company supervisor:

Boudewijn Haas

Company:	Levi9
Position:	Sales & Account Manager
E-mail:	b.haas@levi9.com

December 15th, 2020
University of Twente

ABSTRACT

Topic: This research was conducted under the supervision of the University of Twente and Levi9 Technology Services as a part of a master graduation project. Challenges in software development increasingly revolve around gaining a competitive edge in terms of operational excellence and software quality. It is therefore essential to close the business-IT gap by, breaking down the silo between the two, actively involving both domains in the process, and redistributing responsibility between IT professionals (DevOps), who deliver reliable and stable IT systems, and business professionals, who understand the rationale of IT systems from the business perspective. BizDevOps is the next evolution and the extension of the DevOps chain that aims towards tighter integration of Biz and DevOps stakeholders and processes in order to gear software development and delivery towards the end-user.

Main research question and goal: This thesis paper investigated how to design a robust BizDevOps Method for IT organizations to align their business and IT domains, enhance enterprise agility, deliver software according to business requirements, and maximize end-user experience?

Methodology: Design Science Research Methodology was applied to define the problem, design, and validate the BizDevOps Method based on the insights from a multivocal systematic literature review, semi-structured expert interviews, and focus group evaluation rounds with experts.

Main results: The BizDevOps Method prescribes active commitment of both Biz and IT roles in cross-functional BizDevOps teams that facilitate and drive exploration, development, and validation phases of software development and delivery. The artifact puts a central emphasis on iterative thinking, information flow, the presence of frequent feedback cycles, alignment loops, performance measuring, continuous improvement, and end-user centrism. Literature and practical findings imply the need for cross-functional teams whereby the business stakeholders are active participants capable of bridging the end-user and DevOps, prioritizing tasks and resources to shorten and simplify feedback cycles throughout the whole process. Practices related to continuous feedback are wickedly structured, with multiple qualitative and technical feedback cycles. The key is to integrate and synchronize them all together in order to ensure frequent information flow and progress updates on delivering value in line with the set business goals. Regarding KPI metrics, literature and practice suggest a symbiosis of result-oriented metrics across cultural, performance, and customer domains. This means the prioritization on value-focus, outside-in metrics, and the combined use of leading indicators to aid in business value and goal definition, and lagging indicators for reflecting on performance and value delivery. How these indicators are specified on the unit level is something that has proven to be dependent on the organizational operating sector, business model, and maturity.

Main contributions: The BizDevOps Method with a high degree of abstraction and general applicability is the main contribution of this thesis paper. The artifact guides organization in establishing customer-centric and continuous improvement mechanisms that bridge the gap between DevOps and business goals. This thesis paper is one of the first to combine theoretical and empirical evidence, and conduct validation to provide a holistic and action-based solution to the BizDevOps alignment gap.

Limitations and further research: The findings of this thesis paper are mainly limited by the number of participating organizations. Also, the validation of the artifact was mainly artificial which leaves space for action-based validation research to apply the artifact in a real organization and learn about its effects in practice. It is furthermore recommended that further research focuses on BizDevOps implications for organizational change management, multi-level enterprise governance, and security and risk management.

ACKNOWLEDGEMENTS

After nine long months, in the middle of the global pandemic when everything is at a standstill, the light at the end of the tunnel has been reached. My time at the University of Twente has come to at an end. In the period full of challenges, uncertainty and things changing rapidly day-by-day, I am very satisfied with how this thesis paper has turned out. Nevertheless, without the help of some people, this thesis project would not have been possible. I therefore would like to thank to a variety of people that have participated in this research project.

First of all, I would like to thank Adina Aldea, Olivia Plant, Maria Iacob and Jos van Hillegersberg for being outstanding supervisors for my thesis. I was very luck to have supervisors who have dedicated their time and effort to help me make the most out of my graduation project. Their ideas and feedback at the same time challenged me and motivated me to dive into the core of the topic, literature and methodology from which I was able to exceed my own expectations. Thank you for all the informative and productive meetings that we have had and for your willingness to read my drafts and provide rich input.

I would also like to thank my graduation company Levi9 for giving me the opportunity to do a graduation project with them. Many thanks to my company supervisor, Boudewijn Haas for always finding time to spar with me, provide directions and other forms of support. Another big thank you goes to CCO of Levi9, Debby Jansen and the colleagues from Serbia, Anamarija Petrovic and Damir Solajic for their help, insightful discussions and help in generating interview leads. Levi9 showed trust, openness to ideas and the willingness to help at any time, despite the bottlenecks that the pandemic caused. For that, I am very grateful and could not have wished for a better company to graduate with.

More than 20 people that have taken part in this research as interview respondents and/or validation experts also deserve a big thank you. Without their stories and insights the BizDevOps Method would not have seen the day light. The interesting discussions that I have had with them did not just help me to obtain results for my thesis but have also given me the opportunity to see what are the possible career paths and challenges across the IT sector. Something that will definitely be very useful in my future exploration of career opportunities.

At last, I would like to thank my family and friends for their moral support, who were there when things were not at their finest to lift me up to the ground. I am very happy to have you in my life and I am looking forward to share more moments with you in the future.

TABLE OF CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1 Motivation: operational excellence and user experience

Challenges in software development increasingly revolve around gaining a competitive edge in terms of operational excellence and software quality. Companies are looking for a possibility to speed-up the process, time-to-market, and increase the quality through a better cooperative alignment of business and IT departments (Gruhn & Schäfer, 2015). Silo mentality used to be a common practice occurring with traditional IT approaches. This is the mindset when departments within a company refuse to share information and cooperate with other departments. Such a phenomenon led to a trend of applying agile and lean approaches to maintain the IT-infrastructure. Soon after, IT organizations realized that more value can be created if the collaboration between software development and operations teams was enhanced (Schrader & Droegehorn, 2018).

This gave birth to the DevOps approach that enables the transfer of Agile methods to IT operations and development. As a result, a tighter collaboration of Dev and Ops ensures a minimized risk of untested items, shorter release cycles, and a stable process. Approaches such as DevOps aim at breaking down the silo structures between software development and IT operations to optimize the application development process (Gruhn & Schäfer, 2015). However, it has been noted that DevOps stays within the boundaries of an IT department and does not necessarily address the gap between IT and business. Current approaches for software development optimization are either oriented on the IT side (DevOps) or the business side (end-user software engineering). Hence there is very little attention given to the actual gap between the two. Agile for example looks to improve interaction and communication between business and IT but does not directly look to break the silo, only minimize it. Therefore, there is a need for a new holistic approach that bridges the gap between the IT departments and business departments (Fitzgerald & Stol, 2014; Gruhn & Schäfer, 2015; Drews et al., 2017; Schrader & Droegehorn, 2018; Forbrig, 2018; Wiedemann et al., 2019).

1.2 Problem investigation

The current status quo requires a transition from the domain of business (problem space) to the domain of software engineering (solution space) where the actual software construction takes place. As a result, the business domain has limited participation in the process, being only able to present requirements and review the final software instead of actively participating in the actual software creation (Gruhn & Schäfer, 2015). The separation of business and DevOps function creates a misalignment between what is being built and what the end-user asked. Not having a tight integration of business and DevOps units prevents organizations from properly understanding, communicating, and managing dynamic business requirements. This as a result leads to a deviation from end-user requirements in software development and delivery. BizDevOps is the next evolution and the extension of the chain that addresses this gap by redistributing responsibilities between IT professionals, who deliver reliable and stable IT systems and business departments, who understand the rationale of IT systems from the business perspective (Schrader & Droegehorn, 2018).

BIZ - enables business people to express the requirements in a hands-on manner, thereby reducing the necessary knowledge transfer from business to IT. As such, they are able to accelerate feedback cycles.
DEV - enables IT departments the governance of the application development process that leads to a high quality of software artifacts.

OPS - provides automated tooling and integration that enables development at a high pace (Forbrig, 2018).

Organizations that implement BizDevOps in their approach to software delivery can expect a number of beneficial outcomes. First, BizDevOps organizations can enhance their enterprise agility further, making it possible to quickly respond to changing business needs. Second, BizDevOps assists organizations in aligning its business and IT stakeholders and processes, so that software development and delivery can continuously be geared towards the end-user requirements. In return, a better understanding of the end-user needs enables the BizDevOps organization to minimize product variability, maximizing user experience, and achieve higher revenue (Fitzgerald & Stol, 2014; Gruhn & Schäfer, 2015; Schrader & Droegehorn, 2018; Forbrig, 2018; Wiedemann et al., 2019).

1.2.1 Research goals and objectives

BizDevOps is fairly new and a relatively unexplored concept in the academic literature. Several authors have attempted to add the Biz extension to the DevOps chain with concepts such as continuous planning, continuous innovation, and cross-functional teams coming into the discussion (Fitzgerald & Stol, 2014; Gruhn & Schäfer, 2015; Drews et al., 2017; Schrader & Droegehorn, 2018; Forbrig, 2018; Wiedemann et al., 2019). Nevertheless, few attempts have been made to develop a holistic BizDevOps Method that could formalize and guide the business-IT alignment process while still maintaining a clear sight of the business strategy and goals to be achieved. Therefore, this thesis paper aims to develop a BizDevOps Method, continuously centered on customer requirements and one that promotes active and intensive involvement of the business department in the software development and operations process. The new and holistic BizDevOps Method aims to identify the underlying mechanisms that contribute to the facilitation of the business-IT alignment strategy while ensuring a high degree of enterprise agility. In this context, enterprise agility refers to the ability of an IT organization to adequately and in a timely fashion respond to changing business demands so that it maintains a high degree of operational excellence and end-user experience (Wiedemann et al., 2019).

1.2.1 Stakeholder definition and research scope

This research focuses on examining best software development practices agile IT organizations. The target group of this thesis paper involves business stakeholders and DevOps teams, whereby the interaction between them is examined to see how business (Biz) and IT (DevOps) alignment can be optimized to work on customer-centric software development. It is thereby concerned with how business stakeholders can capture value through product requirement collection and planning for the software development cycle. In addition, the study examines medium and large agile software product organizations that have one or multiple DevOps teams that deliver software components and systems. The nature of the problem in this thesis is addressed through a business-centered scope where people and processes stand central, rather than technological aspects. Therefore, social intelligence, soft skills, and the human-centered software development process are seen as a pre-condition for the advancement of BizDevOps (Forbrig, 2018). Nevertheless, technical components are mentioned in a supplementary role of providing a context to BizDevOps alignment but are not a central focus of the thesis. Other business components such as project portfolio management, financial matters, and budgeting are also out of the scope of this research.

1.3 Research question

This thesis paper focuses on gathering the insights from theory and practice to analyze the requirements, driving mechanisms, and goals in a DevOps-agile software development environment. The new insights will be used to develop a BizDevOps strategy for a business-IT alignment that will enable IT organizations to quickly adapt and deliver on dynamic business needs. For that reason, the main research question of this thesis is formulated as follows:

How can a BizDevOps Method be designed to enhance enterprise agility within an IT organization?

With this question, the main aim of this thesis is to provide IT organizations with a robust method that will allow them to achieve a high degree of operational excellence and end-user experience through a BizDevOps business-IT alignment. Operational excellence and maximized end-user experience imply that the organization needs to have a clear understanding of the business requirements and the capability to align its people, processes, technology, and data towards continuously developing and delivering software with the end-user in mind. To answer the main, purpose-based research question, it is necessary to examine knowledge (KQ) and design (DQ) questions:

KQ1: Which practices to optimize the information flow and shorten the feedback cycles throughout the whole software development process are mentioned in the literature?

KQ2: Which practices to optimize the information flow and shorten the feedback cycles throughout the whole software development process are used in practice?

The aim of KQ1 and KQ 2 is to examine how the communication process throughout the whole software development process takes place. This process involves the cycle from the client to the business product owner (PO) to IT departments and back from the IT department to PO to the client (Forbrig and Herczeg (2015). The optimization of the information flow resulting in shorter feedback cycles is a critical component of business value delivery and thus a necessity to address with KQ1 and KQ2. Namely, shorter feedback cycles allow a BizDevOps team to gain faster feedback from production, customer usage, and possible problems that drive software improvements and higher supportability (Ravichandran et al., 2016). In other words, with optimized information flow and shorter feedback cycles, BizDevOps teams can maximize and accelerate business value delivery (Forbrig, 2018). The main goals of KQ1 and KQ2, in this case, is to acquire insights into practices and drivers of information flows as well as the gaps and challenges that hinder further alignment of business and IT departments to deliver their service optimally. Therefore, to answer the first knowledge question a systematic literature review (SLR) is conducted to gain insights from theory (KQ1). To answer the second knowledge question a qualitative study in the form of interviews with several IT organizations is utilized (KQ2). Among many things, this aspect of the thesis paper will examine how the PO collects and translates the client's requirements to the IT department. Furthermore, the structure and the functioning of agile (Biz)DevOps teams will be examined with the emphasis on business stakeholder involvement in the software development process. The theoretical and practical findings should result in a solid foundation to design a BizDevOps Method that can optimize the information flow and feedback cycle in an IT organization.

KQ3: Which DevOps KPI metrics that communicate and capture business value are mentioned in the literature?

KQ4: Which DevOps KPI metrics that communicate and capture business value are used in practice?

As noted previously, the main use of a formalized BizDevOps Method is to bring about an improvement in the software development and delivery process. Therefore, no single improvement method is complete without a research question addressing performance metrics that can measure progress and goal achievement. Irrespective of the development methodology followed by an IT organization, metrics are an important means for control and continuous improvement of an organization (Korpivaara, 2020). According to Ravichandran et al. (2016), measuring effectiveness in a business context is critical. Consequentially. this aspect of the thesis paper looks to examine the main DevOps key performance indicators (KPIs) and find a way to align these metrics with business goals by looking at best practices from theory (KQ3) and practice (KQ4). In other words, this section aims to identify, transform, and position DevOps KPI's as customer-centric metrics that can be incorporated into the BizDevOps Method.

DQ5: How to design a BizDevOps Method?

This design question addresses the misalignment gap between the business department (Biz) and the IT development and operations department (DevOps). It proposes a solution-based artifact, the BizDevOps Method to bridge this gap. To do so, several modifications to the DevOps processes are introduced to extend the DevOps chain with Biz. As a result, the designed artifact serves as a base and a starting point for IT organizations to apply the extension within their teams and align Business and IT functions to capture more business value. Therefore, this design artifact, the BizDevOps Method is the main contribution of this thesis project. To answer DQ5 the insights from the literature, qualitative study interviews as well as expert opinion are used to build and evaluate the BizDevOps Method.

1.4 Thesis paper outline

The following structure of the thesis is built to execute the research process:

- Chapter 2 provides background information on agile and DevOps software development concepts and practices. These provide basic understanding and foundation for further research in the direction of BizDevOps
- Chapter 3 introduces the design of this research and explains the steps, research methods, and techniques taken to execute the research.
- Chapter 4 describes the steps taken in conducting the systematic literature review and summarizes the academic literature findings.
- Chapter 5 describes the steps taken in conducting the qualitative study and summarizes the findings from semi-structured interviews with (Biz)DevOps experts.
- Chapter 6 integrates the insights from literature and interviews, sets requirement specifications, and introduces the designed BizDevOps Method.
- Chapter 7 summarizes the findings gathered from validation sessions that were used to improve and examine the utility of the BizDevOps Method.
- Chapter 8 discusses the implications and contributions of the research results for scholars and practitioners. It also addresses the limitations, validity, and reliability of the research.
- Chapter 9 sums up the thesis paper, answers the research questions and summarizes the main contributions.

2. AGILE & DEVOPS BACKGROUND

This chapter provides some basic background information on Agile and DevOps approaches. Several definitions, concepts, and key practices are explained to gain a basic understanding and foundation for further research in the direction of BizDevOps.

2.1 Agile software development

Over time software development processes have gradually moved away from heavy IT, built on the assumption that requirements are relatively stable and accordingly the process could be split into stages (e.g. Waterfall). In exchange came light IT which focuses more on flexibility and speed of processes to cope with the turbulent market requirements (Rodríguez et al., 2019). In such a way, traditional, deterministic, and process-oriented approaches have given way to more flexible approaches that emphasize dynamic processes, customer involvement, continuous evolution, and speed. Agile development established itself as the practice of choice in tackling changing business requirements (Ravichandran et al., 2016). It involves iterative problem solving and continuous customer feedback to address business problem complexity. Agile way of working enhanced the ability of organizations to better meeting customer needs, deadlines, budgeting (Top & Demirors, 2019). According to Younas et al. (2018), there are four core values in Agile: people rather than processes and tools, working software rather than documentation, customer collaboration rather than contract negotiation, responsiveness to change rather than following the plan. A visual representation of Agile software development can be found in Figure 2.1.

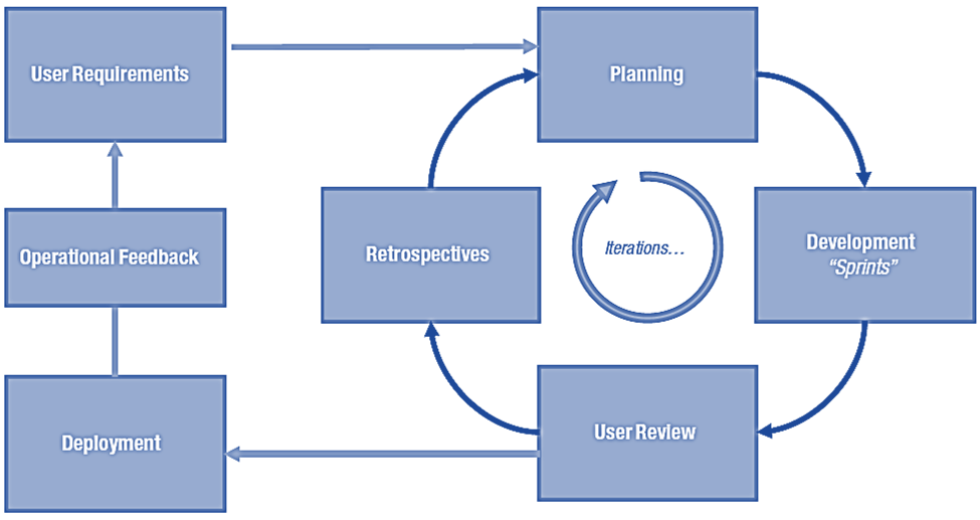

Figure 2.1 - Agile software development model, adapted from Ravichandran et al. (2016, p. 18)

While the Agile approach liberalized the software development process, the structural division between development and operations IT departments remained an issue. Conflicting needs and priorities of the two teams were still causing delays and bottlenecks in the software development process, thereby causing longer release cycles (França et al. 2016).

2.2 Defining DevOps

Many organizations have expressed an increasing interest to break the existing separation between development and operations IT departments. The main goal of bringing the two teams together was to eradicate poor collaboration, time delays, lack of evolvement, and other wasteful behavior that compromises the quality of software development (Lwakatare et al., 2017). DevOps as a combination of two words *development* and *operation* emerged as a viable solution-paradigm that aims for a symbiosis of development and operations to achieve fast release of high-quality software features. (Luz et al., 2019). According to Rodríguez et al. (2019), DevOps extended Agile practices to operations and intensified customer-centrism in software development even further (Figure 2.2). Literature has shown that DevOps is referred to in many different ways such as a framework, methodology, philosophy, mindset, practice, etc. While there is no single universal definition of DevOps in the literature, several general aspects such as culture, collaboration, and automation tools are frequently being mentioned. In the following section, several definitions and understanding of DevOps are presented. Afterward, the thesis expands further on several key DevOps practices.

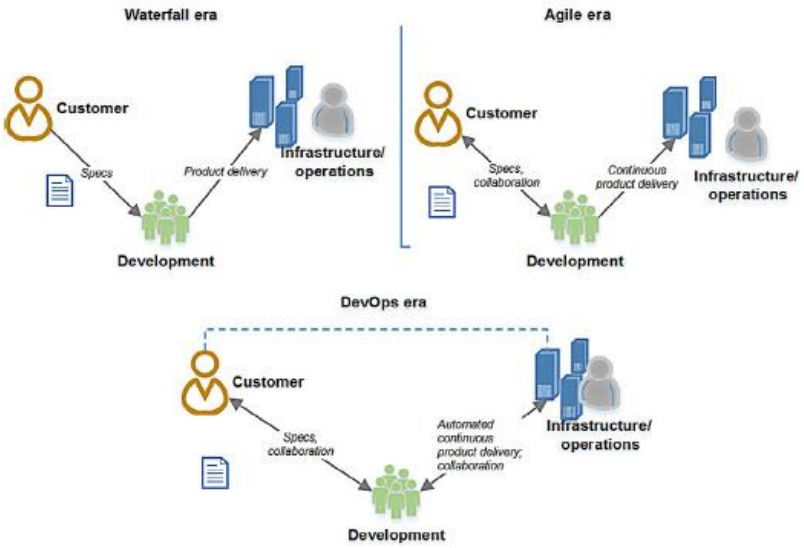

Figure 2.2 – Evolution of software development from Waterfall to Agile to DevOps, adapted from Rodríguez et al. (2019, p.58)

Kaiser (2018) for example, represents main DevOps principles with the acronym CALMS (*Culture-Automation-Lean-Measure-Sharing*). *Culture* in DevOps emphasizes the human component and stresses the importance of establishing collaboration, shared ownership, responsibility, innovation, and experimentation. *Automation* is an essential DevOps principle that enables faster delivery and rapid feedback by automating tasks across the whole development and delivery cycle. *Lean* is another aspect heavily embedded in DevOps practices that emphasizes an efficient way of working and waste elimination. *Measurement* principles guide and monitor processes, thereby serving as quality assurance mechanisms that allow the teams to respond to measured outcomes. Finally, the *Sharing* principle emphasizes information sharing and involving all stakeholders in product development to eliminate the silo mentality. Kaiser (2018) furthermore notes that DevOps is not a standardized framework, but rather a set of good practices which can be grouped into three common elements: *People, Technology, and Process*. Lwakatare (2017) similarly identifies automation, measurement, monitoring, collaboration, and culture as key DevOps principles. Smeds et al. (2015) furthermore provide a set of DevOps

capabilities and cultural and technological enablers represented in Table 2.1. Baškarada et al. (2018) support Smed's vision by framing the definition of DevOps as a combination of practices, tools, and philosophies that aims to unify software development and IT operations. Finally, Jabbari et al. (2016) see DevOps as a development method that with a set of development practices strives to close the gap between development and operations through the emphasis on communication, collaboration, continuous integration, quality assurance, automated deployment, and delivery.

Table 2.1
DevOps capabilities and enablers

Capabilities	• Continuous planning • Collaborative and continuous development • Continuous integration and testing • Capabilities Continuous release and deployment • Continuous infrastructure monitoring and optimization • Continuous user behavior monitoring and feedback • Service failure recovery without delay
Cultural Enablers	• Shared goals, the definition of success, incentives • Shared ways of working, responsibility, collective ownership • Shared values, respect, and trust • Constant, effortless communication • Continuous experimentation and learning
Technological Enablers	• Build automation • Test automation • Deployment automation • Monitoring automation • Recovery automation • Infrastructure automation • Configuration management for code and infrastructure

Note: DevOps capabilities and enablers adapted from Smeds et al. (2015, p. 171)

In the following section, Kaiser's CALMS acronym is used to summarize common DevOps practices. Culture and sharing are combined into a single section and are examined as social and organizational aspects that lead to the establishment of a DevOps culture.

2.2.1 Culture and sharing

People and cultural change are at the heart of the DevOps approach. DevOps strives to create cross-functional teams with common goals and shared responsibility. Kaiser (2018) suggests to put development and operations together and to create channels of communication within the team members. In this context, development and operations collaborate from day one throughout the whole process and collectively participate in the development, monitoring, deployment, and other practices. As a result, the process becomes more organic while a system-thinking mentality is established that the tasks of each team do not start or end at the handover from Dev to Ops. Rather, the work is complete

once the application is successfully delivered to the client in terms of time and quality (Luz et al., 2019). In other words, this creates a sense of shared responsibility and ownership in the teams. By merging the two departments and involving all stakeholders in the production process, information flow and knowledge sharing are promoted. This eliminates competition, skepticism, and silo barriers between development and operations (Kaiser, 2018).

According to Kaiser (2018), a DevOps team must be built around an application and all people responsible for its development and operations need to be brought together. Typically a DevOps team (Figure 2.3) can consist of:

- *Product Owner (PO)* who comes from the business organization and owns the product backlog.
- *Scrum Master (SM)* who leads the development
- *Developer (DEV)* who is responsible for coding and unit testing
- *Tester (TEST)* who develops testing scripts and execute (non)-functional tests
- *Architect (ARC)* who designs the software architecture and is typically shared between teams
- *Database Administrator (DBA)* who manages the database
- *Application support (AS)* who is responsible for application support activities.
- *System administrator (SYS)* who configures and manages tools
- *Service manager (SMG)* who manages services from incidents, problem, change, and other areas
- *IT security (SEC)* who manages security aspects.

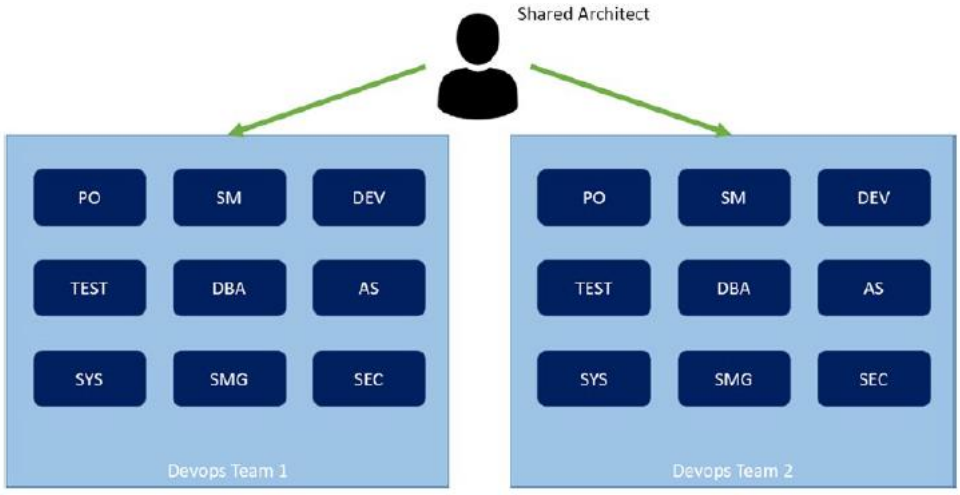

Figure 2.3 – Visualization of a typical DevOps team, adapted from Kaiser (2018, p.20)

2.2.2 Automation

A typical DevOps process involves the following components: plan, code, build, test, release, operate and monitor. Due to the dynamic requirements and increased pace in software development, DevOps works with an automized deployment pipeline where all the steps up to release, including testing are automated. On top of agile software development, three frequently mentioned practices enable DevOps organizations to optimize and automate their processes. The first of these practices is *Continuous Integration* where developers frequently integrate code into a shared source code repository, preferably multiple times per day. Code quality check is then executed through automated tests and builds which ensure quick code conflict resolution and fast delivery (Kaiser, 2018). The second practice frequently mentioned is *Continuous Delivery* which is a set of capabilities that enable code whether that be changes in features, configuration, or bug fixes to be deployed safely, quickly, and sustainably (Forsgren,

Humble & Kim. 2018). After the code has been integrated, a set of automated tests are triggered to validate the code. Finally, *Continuous Deployment* goes a step beyond by completely automating the deployment of the code to production. Whereas in Continuous Delivery is based on a manual trigger, in Continuous Deployment this is triggered automatically after a successful test is executed. Infrastructure as code (*IaC*) is another central concept in DevOps which entails a practice of managing the infrastructure by using scripts to automatically set deployment environments (Rodríguez et al., 2019). The visual representation of the automation processes throughout the development stages can be found in figure Figure 2.4.

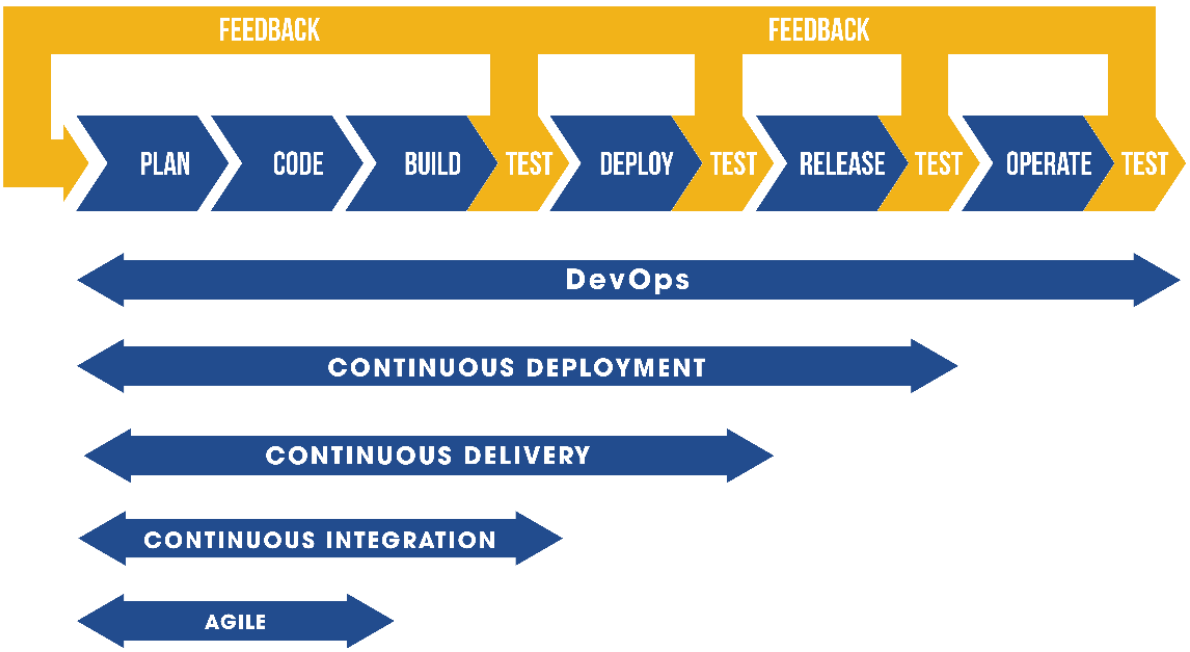

Figure 2.4 – Visualization of the DevOps process automation practices, adapted from LogiGear (2019)

2.2.3 Lean

Another main principle strongly represented in DevOps is the Lean philosophy in software development. Forsgren et al. (2018) identify four main capabilities that make for a Lean approach. The first Lean capability refers to work in small batches where DevOps teams decompose work into features that allow rapid development. In this way, teams develop a prototype, *minimum viable product* (MVP) with just enough features to validate the product with the user. The ability to work in small batches, therefore, allows the teams to quickly collect user feedback by using techniques such as A/B testing. The second Lean capability is actively seeking and implementing customer feedback to inform the design of the application based on features, quality, and customer satisfaction insights. Another important Lean capability noted by Forsgren et al. (2018) is team experimentation that refers to the ability of the teams to try out new ideas and create updates during the development process without requiring approval from external parties. To be effective in experimentation an organization should combine working in small batches, customer feedback, and the last Lean capability referring to a visible workflow. The final Lean capability looks at whether teams have a solid understanding of the workflow throughout the whole process as well as whether the flow is visible to everyone so that product and feature status can be seen (Forsgren et al., 2018).

2.2.4 Monitor and measure

As noted previously quality assurance and resilience are integral when it comes to DevOps. Should an organization decide to automate its software development processes, then a system is needed to provide feedback in case of errors (Kaiser, 2018). Continuous monitoring of application and service in this situation is used in DevOps to ensure the visibility of the success or failures of the system (Lwakatare et al.,2019). Monitoring can be used to detect errors during deployment, to perform a system health check, and to provide DevOps teams with improvement insights based on software usage (Lwakatare et al., 2019; Rodríguez et al., 2019). For the feedback to be possible DevOps organizations need to be capable of distinguishing what is an optimal result. The only way to know this Kaiser (2018) argues is to measure the outcomes. Measurements indicate whether a certain event is categorized as an exception or as a warning. With automation in place, it is therefore extremely important that all critical activities and the supporting infrastructure are monitored and optimized for measurement. Measuring effectiveness according to Ravichandran et al. (2016) should always be conducted in a business context to make the necessary performance and quality adjustments to maximize business value.

2.3 Chapter conclusion

To conclude, this chapter presented background information on the DevOps approach in agile software development. As such, the background information on DevOps serves to establish essential knowledge and understanding of the concepts, need for further exploration of the BizDevOps approach. This chapter started by introducing the paradigm shift from traditional IT approaches to faster and more agile development that could better address customer requirements. DevOps was furthermore introduced as an approach that was adopted to synthesis the development and operations department to break down the silo mentality and improve efficiency in agile software development. Through the CALMS acronym DevOps was defined and explained as a set of collaborative, shared culture, process automation, lean, measuring, and monitoring practices. However, while DevOps seemed to have further liberalized the software development process by closing the gap between different IT departments, a substantial gap between IT and business goals remains to be closed (Fitzgerald & Stol, 2014; Gruhn & Schäfer, 2015; Drews et al., 2017; Schrader & Droegehorn, 2018; Forbrig, 2018; Wiedemann et al., 2019). Therefore, this business-IT gap sets the ground for the extension of DevOps by integrating a business component in the chain, BizDevOps which is addressed more in the following chapters.

3. RESEARCH DESIGN & METHODOLOGY

This chapter provides an overview of the overall research design that this thesis paper utilizes throughout the whole research project.

3.1 Research methodology - Design Science

The research framework that this thesis follows is based on the Design Science methodology by Wieringa (2014). Design Science methodology is a solution-based approach that specifically focuses on design problems within the field of information systems and software engineering. As such, it aims at designing an artifact and investigating the same in a problem context. The studied artifact is designed to interact in a problem context, with the intention to provide improvements in that specific context. To design a proper artifact; in this case the BizDevOps Method, this thesis project includes descriptive research conducted through a systematic literature review and an empirical qualitative study. The insights from this descriptive research aim at answering supplementary knowledge questions (KQ) and design questions (DQ), thereby gaining enough understanding to design an effective and holistic BizDevOps Method. The visualization of the Design Science framework in the BizDevOps context can be found in Figure 3.1 below.

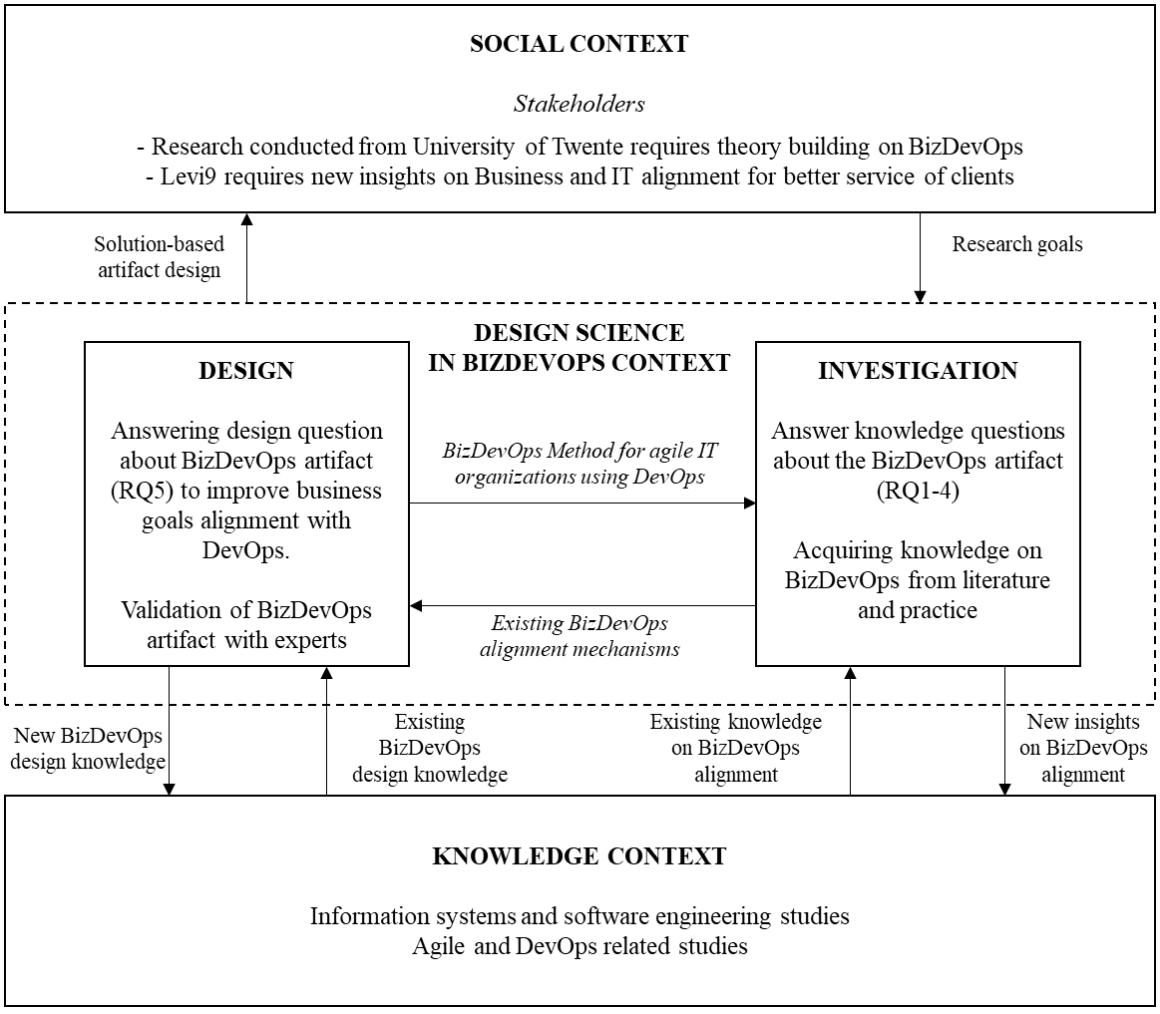

Figure 3.1 – Design science framework in the BizDevOps context, adapted from Wieringa (2014, p. 7)

According to Wieringa (2014), a design research project is guided through a design cycle which this thesis is following. The design cycle encompasses three main stages: 1) *problem investigation*, 2) *treatment design and* 3) *treatment validation*. Figure 3.2 is a visualization of the design cycle applied to the BizDevOps context of the thesis. Each step has a set of corresponding knowledge questions (marked by a question mark) and design problems (marked by an exclamation mark).

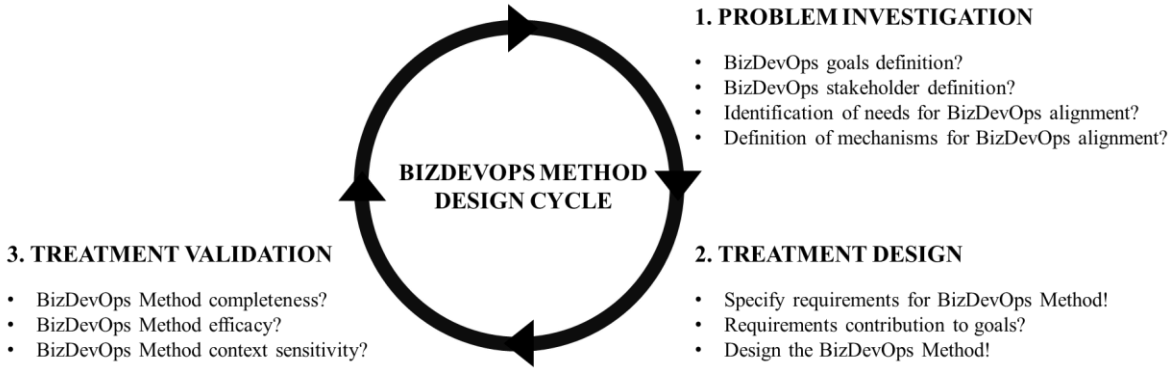

Figure 3.2 – BizDevOps Method design cycle, adapted from Wieringa (2014, p. 28)

3.2 Problem investigation

In the problem investigation stage, the main research goal is to investigate, identify, describe, explain, and evaluate the problem that should be treated. This is done before designing the artifact and setting design requirements for the artifact. Therefore, in the problem investigation stage (covered in chapter 1) this thesis defines the underlying problem, sets main goals and objectives, as well as the corresponding research question.

Design problem:

The business and IT gap hinders companies to achieve a higher degree of operational excellence and high-quality user experience. Previous approaches such as agile and DevOps have optimized the workflow within the IT department but a thorough alignment with business strategy remains to be addressed. Therefore, the extension of DevOps by adding a Business component to it is necessary to address this issue (Fitzgerald & Stol, 2014; Gruhn & Schäfer, 2015; Drews et al., 2017; Schrader & Droegehorn, 2018; Forbrig, 2018; Wiedemann et al., 2019). Hence, we get BizDevOps.

Using the template by Wieringa (2014), the following design problem is developed:

Table 3.1
BizDevOps design problem

Improve <Business-IT alignment>
by <design BizDevOps Method>
that satisfy <enterprise agility>
in order to achieve <operational excellence and user experience>

Note: design problem template adapted from Wieringa (2014, p.16)

3.3 Treatment design

The second stage of the design cycle describes the process taken to design the artifact. The main design artifact of this thesis paper is a BizDevOps Method which aims for the enhancement of enterprise agility through a better alignment of DevOps and business goals. In design science research, a method as an artifact type refers to a set of well-defined activities used to solve a problem and achieve a certain goal (March & Smith, 1995). A method, therefore, has a prescriptive character, meaning that explains what to do in certain situations to arrive at a solution. Many methods are based on underlying constructs (language) and include representational guidelines (model) of the solution space. This includes procedural guidelines, for example, how to work, which steps to take, and what questions to ask (Goldkuhl et al., 1998).

Putting it in the context of this research, the BizDevOps Method aims to define desired steps and describe how they could be performed in the Agile and DevOps software development process. It aims to prescribe rules, guidelines, and behavior patterns which can lead to a better alignment of DevOps with business goals, ultimately allowing IT organizations to maximize their agility and value delivery. Nevertheless, since the BizDevOps Method focusses on strategic alignment of business (Biz) and IT (DevOps) the prescribed set of processes require human cooperation and creativity. Rolland (1998) argues that a method with strategic implications and an emphasis on human interactions should strive to provide flexible guidance rather than strict process enforcement. Goldkuhl et a;. (1998) further argue that a good feature of a method is general applicability, meaning that it is not bound to processes of a single software development company. Therefore, this BizDevOps Method places flexible guidance and general applicability central in providing a solution for a more customer-centric agile software development.

To design the BizDevOps Method several steps are described and explained in chapter 6. Due to a lacking and unestablished empirical research field on the topic of DevOps-Business alignment, both the insights from theory and practices are required for holistic treatment design. As such, the results from the systematic literature review and the qualitative study interviews are used to define the main requirements and challenges that the BizDevOps Method should take into account. Some general requirements which will be elaborated in chapter 6 are that the BizDevOps Method should contain a high level of abstraction. Furthermore, the BizDevOps Method should be holistic, so that it captures the entire software development process and takes all stakeholders into account (Biz, DevOps, and Client). At last, the method should incorporate all requirements and tackle all challenges identified from the systematic literature review and qualitative study interview.

3.4 Treatment validation

According to Venable et al. (2016) validation of the design artifact is a central and critical aspect of design science research. Treatment validation can be defined as a way to justify that the designed artifact in a problem context would contribute to stakeholder goals (Wieringa, 2014). Venable et al. (2016) mention two validation method categories from which different validation strategies are proposed. The first distinction is made between *formative evaluation* and *summative evaluation*. The former strives to improve the characteristics and performance of the artifact based on empirical interpretation. The latter is used to establish a shared understanding of the design artifact in various contexts. The second distinction made by the author is between artificial evaluation and naturalistic evaluation. Artificial evaluation is used to test design hypotheses and includes laboratory experiments and simulations.

Naturalistic evaluation explores the performance of the design artifact in a real context, typically within an organization.

3.4.1 Validation strategy

For the validation of the BizDevOps Method, *Human Risk & Effectiveness* strategy suggested by Venable et al. (2016) will be utilized. This strategy is specifically suitable in situations where the major design risk is social or user-oriented and where it is relatively ease to evaluate the artifact with potential users in a specific context. The critical goal of this evaluation strategy is to rigorously establish that the benefit of the design artifact will be continuous in the real problem-context over the long term. *Human Risk & Effectiveness* evaluation strategy emphasizes (possibly artificial) formative evaluation early in the process but quickly moves on to a more naturalistic formative evaluation. Nearing the end of the process, the strategy focuses on summative evaluation to rigorously examine the effectiveness of the artifact. However, it should be noted that the extent of naturalistic evaluation applied in this thesis paper is limited by time constraints as well as the pandemic, which is why the real implementation of the artifact was not possible. Yet, as an alternative, this thesis paper took the effort to simulate the setting of naturalistic evaluation. All respondents were asked to imagine how the artifact could be implemented and placed in in a real setting while evaluating it.

In the context of this research, formative evaluation is executed through expert opinion where a discussion was held with several (Biz)DevOps and agile experts. The insights collected from the experts served as a learning activity that helped in the improvement of the designed artifact, the BizDevOps Method. After the iterations to the BizDevOps Method were made, the improved and final version of the artifact was presented to the field experts as a part of the summative evaluation. In the summative evaluation stage, the experts placed the BizDevOps Method in the context of their organizations and provided feedback on understandability, completeness & accuracy, usability & efficacy, and organizational fit of the final version of the artifact. In other words, they examined to what extent did the BizDevOps Method attain the goal for which it was designed; to align DevOps with business goals for the enhancement of enterprise agility and high-quality software delivery.

3.5 Research methods and techniques

The following section summarizes the research methods and techniques used for the execution of this design science research. The details on the execution, steps, processes, and results of the methods can be found in the chapters corresponding to each method and technique used.

3.5.1 Systematic literature review

The first data collection method used to answer knowledge questions KQ1 and KQ3 is a systematic literature review. According to Kitchenham (2004), a systematic literature review is a secondary study that is used as a means to identify, evaluate, and interpret all available research relevant to a topic of interest. The author identifies three stages for conducting a systematic literature review: *planning*, *execution*, and *result analysis*. In the planning stage, a research strategy and a research process for the systematic literature review have been set to minimize bias and maximize consistency in the selection of the (Biz)DevOps literature. Based on exploratory research a set of relevant keywords have been identified and were used in the execution stage to search for relevant papers across several academic databases. All papers found, went through a selection process in which inclusion/exclusion criteria were defined. Finally, thematic categorized of the remaining papers was used to structure and present the

results of the systematic literature review. Chapter 4 provides a detailed explanation of the systematic literature review process from planning steps to execution and result analysis.

3.5.2 Qualitative study interviews

The second data collection method used to answer knowledge questions KQ2 and KQ4 is qualitative study interviews. Oates (2005) defines an interview as a specific conversation that is led by a researcher, generally follows an agenda and it has a set of unspoken assumptions. For this research, semi-structured interviews will be conducted with Agile and (Biz)DevOps experts from several IT organizations. A semi-structured interview is an interview type where the interviewer asks a pre-determined set of questions, is involved in a conversation, and may ask additional questions, change the order or content based on the conversation flow (Oates, 2005). The main goal of this empirical qualitative study is to understand how different companies scale DevOps and align it with business strategy to capture business value. This includes analyzing the practices and processes of the companies and incorporating the findings into the BizDevOps Method.

The interviews were transcribed and analyzed, following a mixed principle of deductive and inductive coding described by Seale (2004). A deductive approach was taken to identify some themes and develop codes before the interviews. The literature and research questions were used as the basis for developing a codebook. During the analysis, a more inductive approach was taken to create and add new codes. *Open coding* was applied to chunks of text to generate concepts and categories. These categories served to identify high-level practices that are essential in aligning DevOps with business goals. Furthermore, *axial coding* was applied to define relationships between codes and develop broader conceptual categories. For the qualitative study analysis and results refer to chapter 5.

3.5.3 Validation method - expert opinion focus groups

This thesis paper relies on expert opinion to validate the designed artifact, the BizDevOps Method. Expert opinion will be collected using focus group sessions. A Focus group can be defined as a moderated discussion among a small group (up to 12 participants) on a certain topic. In the formative evaluation round, two exploratory focus group session have been organized. According to Tremblay et al. (2010), an *exploratory focus group* in a software development setting is used to achieve rapid incremental improvements in the designed artifact. After the refinement of the BizDevOps Method, all participants have received the refined version of the artifact, together with a feedback form, containing criteria-specific questions on a scale 1-5. Therefore, this evaluation form makes a part of the summative evaluation round which strives to confirm the artifact utility, accuracy, understandability and organizational fit.

Regarding the organizing aspects of the focus group sessions, all participants have received informative materials before the sessions. Each participant has beforehand received the description and visualization of the designed artifact, the BizDevOps Method. Additionally, presentation slides, agenda discussion points, as well as evaluation forms with questions, have also been provided, so that the participants could prepare. During the session, a presentation held to introduce and elaborate on the BizDevOps Method. Afterward, the session moved to a critical discussion segment where the participants were able to provide their feedback based on the criteria in the evaluation form. The evaluation form also included a visualization of the BizDevOps Method, so that the participants could also indicate changes in a more visual way. Refer to chapter 7 for more details on the validation process and findings.

3.6 Research model for the BizDevOps Method

Due to the lack of empirical research related to DevOps and alignment to business goals, data has been collected through a systematic literature review as well as through a qualitative study consisting of several interviews with agile IT organizations. The insights gathered from these studies will be incorporated into a method that serves as a guideline for aligning DevOps with business goals, thereby allowing IT organizations to capture more value, provide operational excellence and end-user experience. The BizDevOps Method draft will be presented to interviewed field experts and a focus group session will be conducted to evaluate the method's applicability in a real-life context. The input from these sessions will be synthesized to create the final BizDevOps Method. The resulting research questions and the research design approach are presented in a research model (Figure 3.3) annotated according to the work of Verchuren and Doorewaard (2015). The arrows at the bottom of the figure present the research stage of the design cycle.

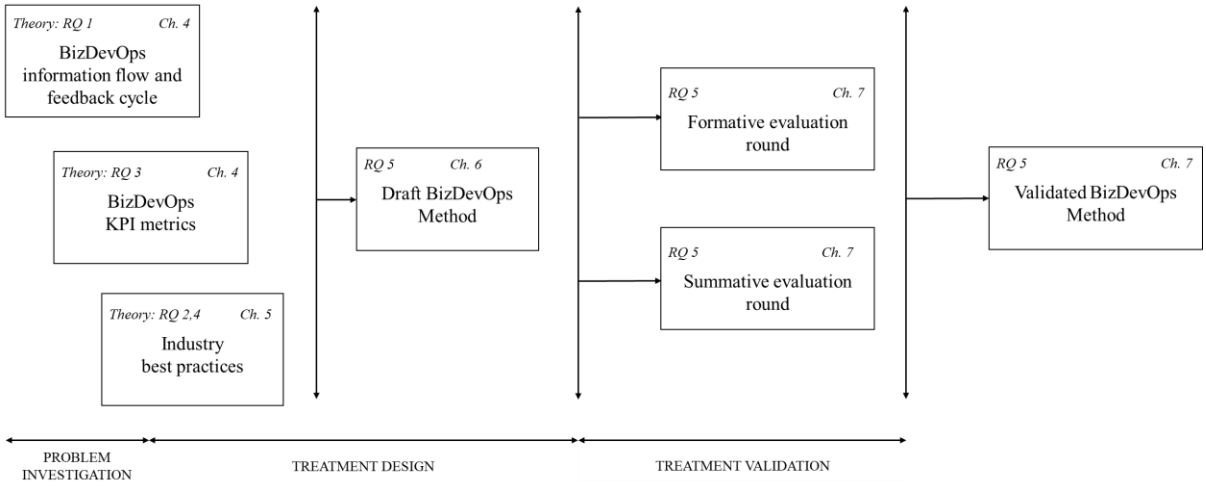

Figure 3.3 – Visualization of the research model according to the notation of Verschuren and Doorewaard (2015).

3.7 Chapter conclusion

To sum it up, this chapter presented the research design and research methods used for this thesis paper. Following the Design Science Methodology from Wieringa (2014), this thesis paper aims to develop a solution-based artifact for a better alignment of DevOps with business goals. After defining the main goals and research questions in the problem definition stage, in the treatment design stage, it was explained that the main goal of this is to design a BizDevOps Method. This artifact aims to define desired steps, provide guidelines and behavior patterns that can lead to a better alignment of DevOps with business goals, ultimately allowing IT organizations to maximize their agility and value delivery. Systematic literature review and qualitative study semi-structured interviews are used as data collection methods to answer all relevant knowledge questions that are necessary before designing a BizDevOps Method. The insights gained from theory and practice serve as input for the BizDevOps Method requirement specification and ultimately the design of the artifact. The requirements for the BizDevOps Method were on purpose specified and refined only after the thorough literature and practical study, which provided solid insights into the most important challenges, problems and priorities that organizations face in meeting business requirements. Furthermore, this chapter provided the

information on the validation of the BizDevOps Method and showed that the artifact is subjected to both formative and summative validation by field experts. A focus group session is used as the main research method to collect data in the validation stage of this research. At last, accordingly, to the aforementioned information, a research model visualization for design a BizDevOps Method has been presented in Figure 3.3. A more detailed overview summarizing the three stages of the design cycle, research questions, and corresponding research methods can be found in Table 3.2.

Table 3.2
Design Cycle for BizDevOps Method

RESEARCH STAGE	RESEARCH METHODOLOGY	CHAPTER
Problem investigation		
Identifying needs for BizDevOps	Systematic literature review	1 & 4
Defining BizDevOps, goals, and stakeholders	Systematic literature review	1 & 4
Defining mechanisms for BizDevOps alignment	Systematic literature review	1 & 4
Treatment design		
KQ1: Which practices to optimize the information flow and shorten the feedback cycles throughout the whole software development process are mentioned in the literature?	Systematic literature review	4
KQ3: Which DevOps KPI metrics that communicate and capture business value are mentioned in the literature?	Systematic literature review	4
KQ2: Which practices to optimize the information flow and shorten the feedback cycles throughout the whole software development process are used in practice?	Qualitative study interviews	5
KQ4: Which DevOps KPI metrics that communicate and capture business value are used in practice?	Qualitative study interviews	5
DQ5: How to design a BizDevOps Method?	Artifact design	6
Treatment validation		
Formative evaluation: artifact improvement	Expert opinion (Exploratory focus group)	7
Summative evaluation: artifact applicability	Expert opinion (Evaluation form)	7

4. SYSTEMATIC LITERATURE REVIEW

This chapter presents the results of the literature review surrounding BizDevOps and related concepts. Following the principles and insights of Kitchenham (2004) and Rouhani et. al (2015), a systematic literature review has been conducted. Both of these academic works specifically focus on conducting a systematic literature review within the field of information and software engineering which is why they were taken as the guideline for this thesis. In the planning stage, a systematic literature review protocol has been created which can be found in Appendix A. It includes all the steps taken to conduct a literature review including the goals of the systematic literature review, search terms, selection criteria, selection process, and paper classification.

4.1 Systematic literature review process description

The following figure visualizes the approach to conducting the systematic literature review.

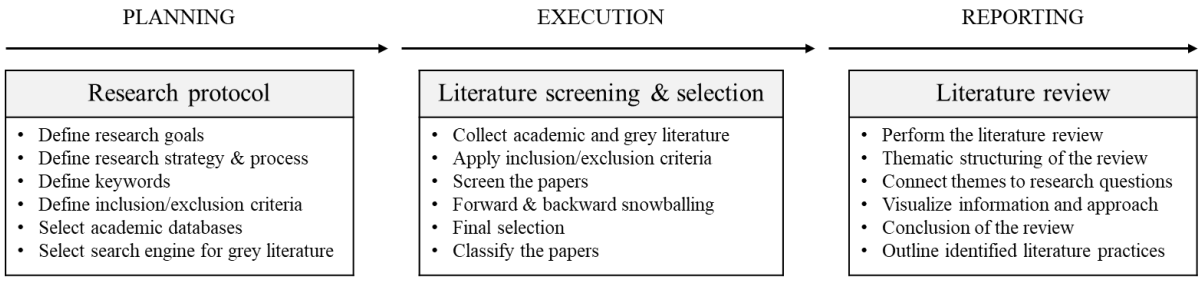

Figure 4.1 – Systematic literature review approach

4.1.1 Planning stage

A research strategy and a research process for the systematic literature review have been set to minimize bias and maximize consistency in the selection of the (Biz)DevOps literature. The first step of the structured process included the setting of research goals for the literature review. In the context of BizDevOps and enterprise agility, the main goals of the literature review were to summarize the existing base of knowledge, identify gaps in the current body of literature and to appropriately position new research activities.

The next stage of the research strategy included the selection of several prominent academic databases such as Scopus, ACM Digital Library, IEEE Xplore, and AIS e-Library. The reason for specifically choosing these databases is based on the recommendations of scholars, peers, previous experience in using these databases as well the amount of relevant and related work to the topic of Agile and (Biz)DevOps. Based on exploratory research, several keyword combinations related to BizDevOps and enterprise agility have been formed and Boolean syntax has been used for searching through the academic libraries. Additionally, inclusion and exclusion criteria (Table 4.1) have been set for filtering the relevancy of the search results. One such criterion was to only include journal papers, conference papers, books, thesis papers and whitepapers published in the range of 2015-2020 to ensure the timeliness and prevent obsolete data from entering the thesis.

Search key: *(“BizDev” OR “BizDevOps” OR “DevOps") AND ("enterprise agility" OR "business process" OR "alignment" OR “metrics”)*

Table 4.1
Inclusion/exclusion criteria for the systematic literature review

INCLUDE	EXCLUDE
Papers about software engineering in English	Papers that only conceptualize Agile methods
Papers about (Biz)DevOps and IT alignment	Papers that only conceptualize DevOps
Papers about (Biz)DevOps and enterprise agility	Papers that only cover (Biz)DevOps automation tools
Papers about (Biz)DevOps and business processes	Papers outside of the topic and not in English
Papers about (Biz)DevOps and business KPI's.	

4.1.2 Execution stage

The screening process of the systematic literature review was further conducted by filtering through the titles of the papers. The narrowing down continued by reading the abstracts of the remaining papers to determining whether there are any relevant grounds to keep them under consideration. Finally, the last papers remaining on the list were fully read after which the final selection was made. Nevertheless, to assure that important academic work was not missed, backward/forward snowball technique suggested by Wohlin (2014) was applied to search examine the references of the selected papers as well as related works that cited the selected papers. The same inclusion/exclusion that were defined previously were used for the selection process. Several relevant titles published before 2015 were found. Yet, regardless of this fact, they have been taken into the literature review either as an independent paper or as a supplement to build on the underlying context of other BizDevOps literature.

To ensure for quality and credibility of literature sources, the initial intention of this thesis paper was only to consider peer-reviewed journal articles and conference papers from academic databases. However, seeing the limited number of academic publications on BizDevOps, grey literature was also taken into consideration. According to Garousi et al. (2016), a systematic literature review in software engineering may not provide a complete insight into the state of practice if only formal literature is examined. As a result, a Multivocal Literature Review which also includes non-peer-reviewed literature can be a good complement to the formal literature. Therefore, this thesis paper also considered grey literature with the special attention given to the quality of the sources. Only first-tier grey literature was considered such as books, whitepapers, and thesis papers. The exploratory research conducted through Google Scholar resulted in three additional grey literature sources. The visualization of the whole systematic review process including the grey literature can be found in Figure 4.2 and it resulted in 19 sources.

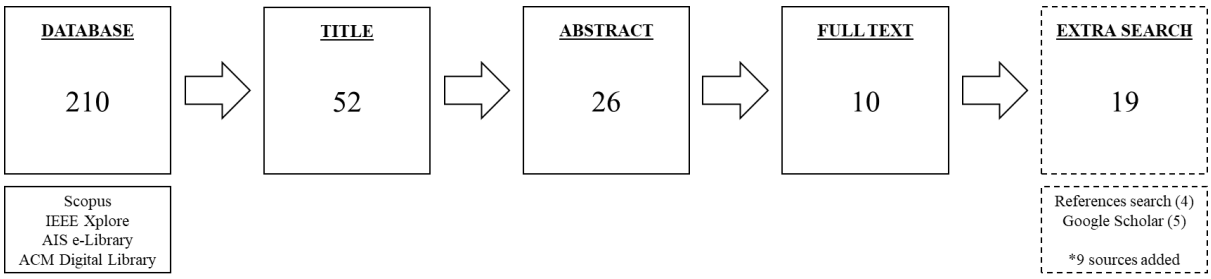

Figure 4.2 – Systematic literature review process and results

4.2 Reporting stage: classification and thematic structuring

From the selected papers, the majority of them can be classified as conference papers, with the remaining papers being journal articles, books, thesis papers, and white paper. If the selected papers are examined (Appendix B), the first thing to note is that all of them are very recent, with the oldest publication being in 2014. Considering that BizDevOps is an extension to DevOps which itself is also a young research field that gained ground from 2010 onwards, it comes as no surprise that the BizDevOps as a field of study is not established. The majority of the papers analyzed focused on defining and conceptualizing the components that make BizDevOps. Only a few papers base their findings on empirical research such as case studies and interviews. There are furthermore even fewer papers that focus on validation and solution-based research. Several authors present frameworks and models for BizDevOps but those are rather conceptual. As a result, this provides good implications for the need for a solution-based research approach in the BizDevOps field which this thesis paper is focusing on.

To structure the systematic literature review a thematic categorization of the selected papers has been used. The thematic categories were developed based on their frequency of discussion in the papers, convergence, and divergence in academic opinion and based on the research questions of this thesis paper. The following seven thematic categories have been identified in Table 4.2.

Table 4.2
Thematic categorization of the systematic literature review

THEMATIC CATEGORY	DESCRIPTION	RESEARCH QUESTION
4.3 Defining BizDevOps	Definition, conceptualization, and discussion on BizDevOps understanding.	…
4.4 BizDevOps cross-functional teams	Examination of organizational aspects and team structure in BizDevOps	KQ1
4.5 The role of Biz in BizDevOps	Examination of roles and contribution of business stakeholders in BizDevOps	KQ1
4.6 Continuous in BizDevOps	Examination of continuous software engineering practices in BizDevOps	KQ1
4.7 Subject-oriented approach in BizDevOps	Expansion on continuous requirement engineering practices in BizDevOps	KQ1
4.8 BizDevOps and continuous BPM improvement	Expansion on continuous improvement practices in BizDevOps	KQ1 & KQ3
4.9 Capturing value with BizDevOps metrics	Examination of BizDevOps KPI metrics that capture business value	KQ3

4.3 Defining BizDevOps

In the earliest publication in this literature review, Fitzgerald and Stol (2014) claim to have coined the term BizDev which they see as a stand-alone entity that complements DevOps. As such Fitzgerald and Stol (2014) see it as continuity between business strategy and software development team. BizDev should, therefore, complement DevOps to achieve continuous delivery and innovation. In addition, citing Fitzgerald and Stol (2014), Moreira and De Franca (2019) use the term BizDev to describe the

upstream collaboration between business strategy and development and DevOps as a downstream collaboration between development and operations. On the other hand, there are a few publications by Forbrig (2018) and Wiedemann et al. (2019) that specifically address the difference between BizDev and BizDevOps. In contrast to Fitzgerald, Stol (2014), Moreira and De Franca (2019), Forbrig (2018) and Wiedemann et al. (2019) see BizDevOps as an extension to DevOps and therefore as one holistic concept. Gruhn and Schäfer (2015) add to this holistic notion by referring to BizDevOps as an approach that strives to narrow the gap between the business department and the IT department within a company, not just development. This is a problem that according to Gruhn and Schäfer (2015) has not been addressed so far by other approaches such as DevOps, agile, and End-User Software Engineering. Forbrig (2018) furthermore adds that the idea of BizDevOps is to address this problem encouraging business, development, and operations staff to work together so that the organization can develop software faster, be more responsive to the end-user demand and ultimately maximize revenue. By adding the Biz extension to the DevOps chain the organization can capture more value with business stakeholder-focus on innovation and benefit. The visual representation of the BizDevOps chain is found in Figure 4.3.

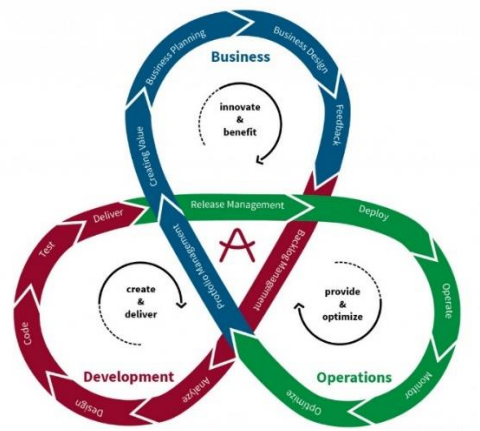

Figure 4.3 – Visualization of the BizDevOps chain, adapted from Forbrig (2018, p.4)

4.4 BizDevOps cross-functional teams

Some publications emphasized the organizational aspects such as the creation of multidisciplinary teams from the domain of Business and IT. These publications see the idea of cross-functional teams as a central notion in BizDevOps (Schrader & Droegehorn, 2018; Drews et al., 2017; Wiedemann et al., 2019). In this sense, Schrader and Droegehoern (2018), Drews et. al (2017) and Wiedemann et al. (2019) do not contradict BizDevOps definition from other authors but rather build further and refine it by upscaling departments to close the gap as one cross-functional team. The main idea in this context is to have a cross-functional team that is responsible for the product or a piece of the product from the ideation stage up until the delivery to the client as well as operations of the product after development (Schrader & Droegehorn, 2018; Drews et al., 2017; Wiedemann et al., 2019).

According to Schrader and Droegehorn (2018), business people do not only set the requirements but are an active member of the team that set priorities for agile software development sprints and backlogs together with developers. BizDevOps makes it possible to restore the innovative power of IT to business departments, thereby jointly creating optimal solutions. Furthermore, BizDevOps should be used for the efficient realization of the digitization strategy. IT is essential in cases where software is the core of the business and where the organizational culture supports the approach. In addition, BizDevOps

follows the agile manifesto, thereby giving preference to individuals, interaction, working software, client collaboration, and responding to change. In cross-functional teams, business people can gain a deeper understanding of new technologies and trends, learn from team members how to exploit them. In other words, the newly gained insights from the business department can lead to the development of new business models with highly innovative products and services. The visualization of the traditional organization of teams and BizDevOps teams is found in Figure 4.4.

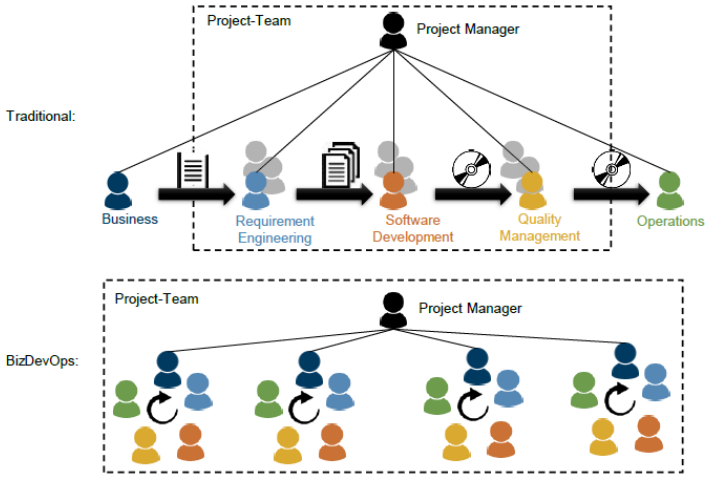

Figure 4.4 – Visualization of the difference between traditional teams and BizDevOps teams, adapted from Schrader and Drogehorn (2018, p.91)

Drews et al. (2017) also emphasized cross-functional BizDevOps teams to address the business-IT gap. The author presents a case study in which 10 BizDevOps teams are working on developing an e-commerce platform. The teams plan, build, and operate an architecture based on micro-services. First, each team has a business designer, a person from the business domain who defines what will be developed in the following iterations. Second, each team has developers that utilize agile-scrum mode to build, test, and run parts of the e-commerce platform. Finally, the technical architects coordinate the operational issues across the architecture. Next to these roles, there is one central role in the department that concentrates on orchestrating and supporting the activities of the teams. In this way, Drews et al. (2017) outline four basic dedication parts of the teams to ensure the platform's functionality, including 1) navigation & search, 2) product details, 3) recommendation and 4) order processing. In this new enterprise architecture management setting, the teams have a high degree of autonomy provide recommendations instead of guidelines on certain technology use, while at the same time allowing for scalable and fluid architecture. The visualization of the platform operated by BizDevOps teams can be found in Figure 4.5.

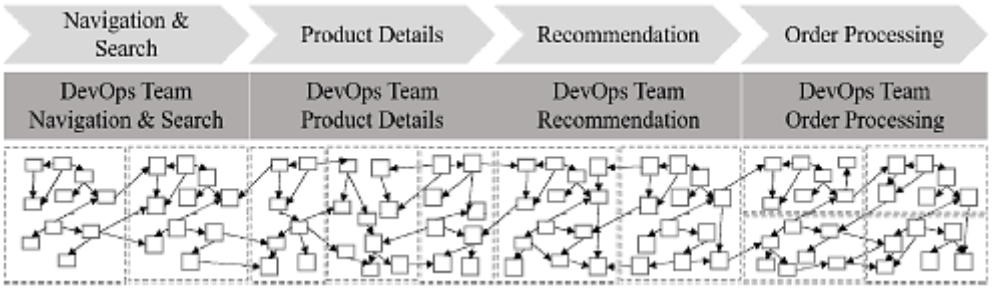

Figure 4.5 - Visualization of a platform operated by BizDevOps teams, adapted from Drews et al. (2017, p.60)

4.5 The role of “Biz” in BizDevOps

Authors such as Gruhn and Shäfer (2015), Moreira and De Franca (2019) go even further in the organizational aspect of the BizDevOps setting by thoroughly examining the role of the business department in the BizDevOps chain. Gruhn and Shäfer (2015) introduce a sandbox Therefore, the authors introduce a sandbox platform that facilitates the involvement of the business stakeholders and allows them to gain technical skills to conduct some software engineering themselves. Following this logic, certain business units should be able to perform a low-code environment; so-called shadow IT to develop data models, user application, and interface. The IT department safeguards the application development process outcome and ensures appropriate quality control. This is done by having software components under full control (managed resources) and putting them under an IT compliance framework. IT department thus safeguards how parts developed by the business department interact with the IT system landscape. Forbrig (2018) furthermore comments on the platform, arguing that this approach allows the business department to express and review requirements through active participation. As a result, Forbrig (2018) argues that this reduces the necessary knowledge transfer from business to IT, thereby ensuring the fastest possible feedback cycle (the “Biz” in DevOps). Additionally, the BizDevops approach allows the IT department to govern the application process by ensuring high quality (the “Dev” in BizDevOps) and to provide automated toolchain integration to speed up the development pace (the “Ops” in BizDevOps). Nevertheless, Forbrig (2018) notes that while the sandbox platform is a good base for the BizDev part of the chain, the DevOps tool support is missing in the platform. Forbrig (2018) builds further on this notion in with his S-BPM which will be covered later on in this literature review. The visualization of a BizDevOps platform can be found in Figure 4.6.

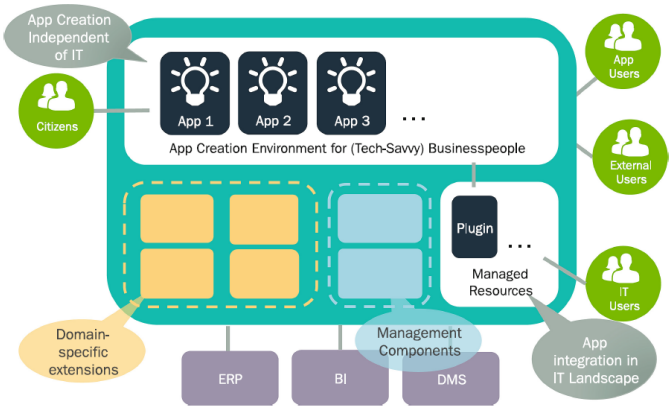

Figure 4.6 – Visualization of the elements of a BizDevOps platform, adapted from Gruhn and Shäfer (2015, p.393)

Next to the sandbox platform by Gruhn and Shäfer (2015), Moreira and De Franca (2019) also address the involvement of business stakeholders in the software development process arguing that the composition of the business team may include people focused on content, requirements, administration and even the customer. In this way, the business area behaves as a customer for the IT area and establishes demands, specifications, and deadlines. This process can also take place with the participation of the IT department. Moreira and De Franca (2019) identify three essential functions of the business department within the process: 1) business tracker, 2) business ambassador, and 3) business expert.

First, the Business Tracker is responsible for the activities related to the creation and continuous maintenance of the requirement list. This includes user stories, product backlogs, and feature list. In this way, the Business Tracker guarantees that the software development process is guided by the requirement list. It is the Product Owner (PO) who is usually responsible on this front argue the authors. Second, the Business Ambassador consists of all activities related to the translation of user needs into business requirements and later on into features to be developed. Additional activities include negotiating deadlines with the customer. The Business Ambassador is expected to have the ability to explain technical constraints and the needed resources for the client to understand the demands needed to fulfil each deadline. The ambassador should also inform developers of requirement priorities so that optimal effort can be invested in maximizing value delivered. The responsibilities of the Business Ambassador also correspond to the PO.

The final role, the Business Expert does not consist of well-defined tasks, argue Moreira and De Franca (2019). It is rather defined by a single responsibility which is having deep knowledge of the business domain which makes the business expert a source of requirements. Business Experts should be available during different stages of the project to reduce the time it takes to get additional information and specification of requirements. The authors note that it is usually the customer that works on this end of the spectrum and introduce the notion of *Customer-on-site*. The notion of *Customer-on-site* entails that the customer is more available and works closely with the IT department. The authors argue that in this way, the *on-site-customer* can see difficulties in that the IT department might have in understanding the requirements written by the business department. The customer is then able to eliminate these bottlenecks by providing additional insights. This according to Moreira and De Franca (2019) increases productivity due to fewer interruptions and requirement specification being needed. The *on-site-customer* in this sense acts as a valuable source of business knowledge and can reduce feedback cycles.

4.6 Continuous in BizDevOps

Continuous Integration, Continuous Deployment, and Continuous Delivery are essential practices used in DevOps to optimize the collaboration between Development and Operations departments. Fitzgerald and Stol (2017) build further on the continuous umbrella concept through Continuous Engineering practices, thereby addressing the gap between Business with Development and Operations. Fitzgerald and Stol (2017) provide continuous software engineering (CSE) model that considers ‘continuous’ as a holistic endeavor. This means that the model considers the entire software life-cycle and takes three main subphases; namely business strategy, development, and operations. Practices such as continuous planning, continuous improvement, and continuous experimentation and innovation, in this case, are seen as the bridges that address the misalignment of business strategy with IT.

Continuous planning and budgeting in this sense represent a holistic effort to involve multiple stakeholders from business and IT functions. The plans are seen as dynamic and evolving open-ended artifacts that respond and are adjusted in accordance with the changing business environment. Budgeting in this context is a continuous activity that facilitates the changes during the year rather than being traditionally an annual event (Fitzgerald & Stol, 2017). Wiedemann et al. (2019) furthermore build on the notion of continuous planning, thereby outlining scalability, security, and quality as key mechanisms of continuous planning. In traditional software development projects, customers could plan their requirements for very long release cycles. Thus, introducing change during the process was difficult because the planning process already closed. This problem can be avoided with the

implementation of BizDevOps continuous planning where the introduction of changes, requirements, and ideas is possible at all times (Wiedemann et al., 2019).

Continuous improvement is established on lean principles of waste elimination and data-driven decision-making that result in small incremental quality improvement which can give an organization a competitive edge and a variety of benefits. *Continuous innovation* is a sustainable process that supports responsiveness to new requirements and changing market demands throughout the software development process It strives to enable processes that help in responding to new market conditions. Integration of Product Owner within DevOps setting is essential for the achievement of continuous innovation (Wiedemann et al. 2019). Continuous innovation is seen as a sustainable and responsible process that is based on appropriate metrics that cover the entire lifecycle of planning, development, and run-time operations. The CSE model also includes continuous experimentation with stakeholders consisting of build-measure-learn cycles that are repeated (Fitzgerald & Stol, 2017). The visualization of the CSE model can be found in Figure 4.7.

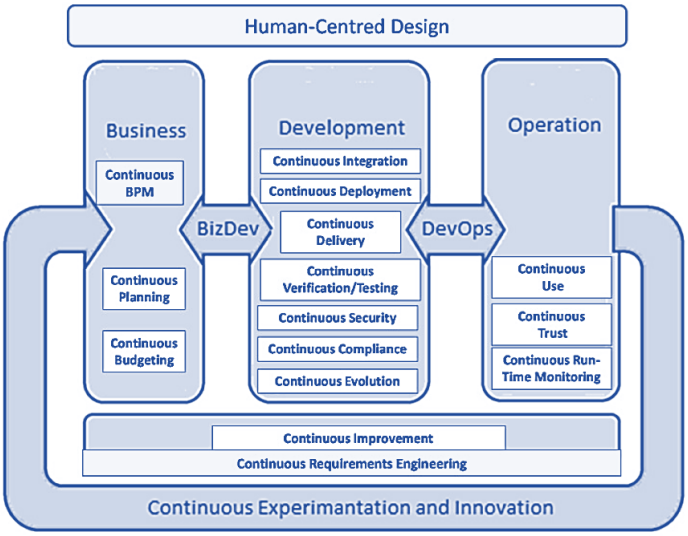

Figure 4.7 – Visualization of the CSE model, adjusted from Fitzgerald and Stol (2017, p.6) and Forbrig (2017, p.8)

4.7 Subject-oriented approach in BizDevOps

Forbrig and Dittmar (2018) expand further on the CSE model from Fitzgerald and Stol (2017) with their Subject-oriented approach in BizDevOps. The authors propose three new additions to the model: Continuous Subject-oriented Business Process Modeling (S-BPM), Continuous Human-Centered Design, and Continuous Requirements Engineering.

S-BPM looks into how Continuous Business Process Modeling (BPM) enables BizDevOps. BPM in this way gives stakeholder groups a reference point and a common language as a guide in their decision-making process. Following the logic that the business department should be involved in the software development process, Forbrig (2017) utilizes S-BPM which is a subject-oriented business process modeling approach that enables business stakeholders to specify workflows, processes and translate them into code using domain-specific knowledge. As such, domain (business) experts can be actively involved in the software development process (Forbrig, 2018).

Continuous Human-Centered Design (HCD) displayed in Figure 4.8 conceptualizes BizDevOps term from a requirement engineering perspective (Forbrig, 2017). In this case, requirements are seen as a starting point of the software development life cycle, where business stakeholders are the main source in gathering these requirements. He brings up storytelling as a potential means of communication between development and business. Storytelling, in this case, is defined as a technique that supports managers in communicating the company's vision, values, and culture across the whole organization. In the BizDevOps context, this approach can facilitate cooperation and communication between the management, business analysts, and development (Forbrig & Dittmar, 2018).

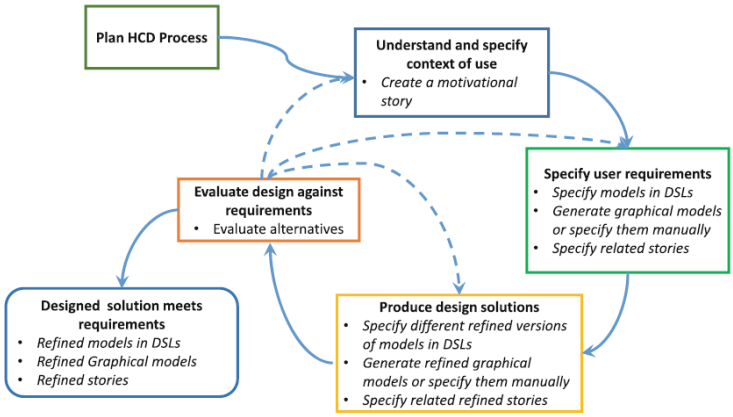

Figure 4.8 – Suggested Human-Centered Design process for functional requirements in BizDevOps, adjusted from Forbrig and Dittmar (2018, p.333)

Furthermore, Forbrig and Herczeg (2015) expand further on the notion of HCD by integrating it into agile development. The authors, in this case, apply the HCD to Scrum (Figure 4.9), but argue for the general applicability of the concept for other agile approaches. The process starts with a Zero-Sprint (from vision to needs), where a business analyst or possibly a developer analyses the context to identify the basic needs of a project. The collection and identification of requirements take place in the following sprint (from needs to product backlog) and it follows an HCD approach. In the next stage (sprint backlog to product) development sprint and requirement sprint are performed parallelly and are iterated in cycles until the end of the project.

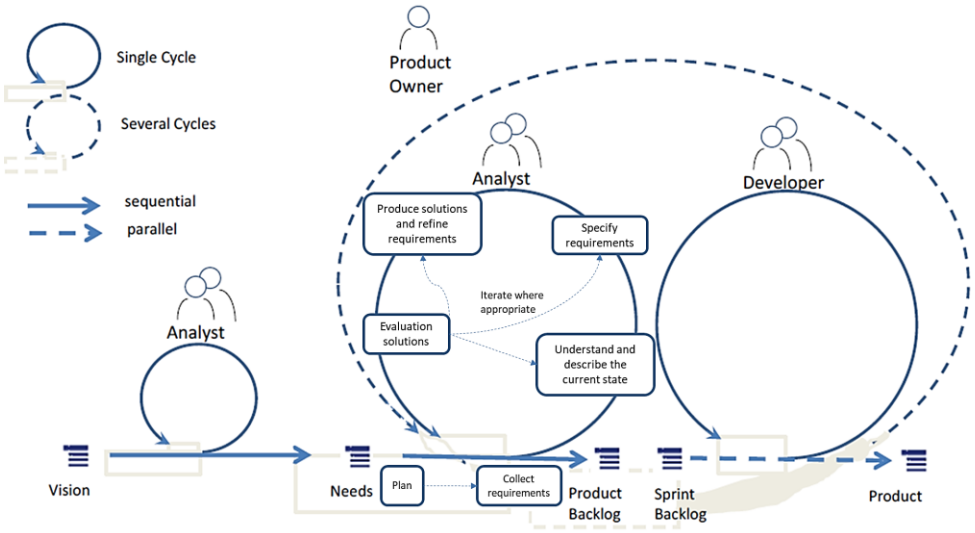

Figure 4.9 - Suggested Human-Centered Design for Scrum, adapted from Forbrig and Herczeg (2015,p. 7)

Lean UX proposed by Elberzhager et al. (2017) is another user-centered approach that combines three development methodologies for rapid feedback collection: design thinking, agile development, and Lean Start-up. Design thinking is a non-linear and iterative process that teams apply to understand users, challenges, redefine problems, develop innovative solutions for prototyping and testing. It is a multi-disciplinary method that also encourages non-designers to use as it supports collaborative design across different roles. Elberzhager et al. (2017) note that next to being customer-centric, design thinking methodology also takes technical capabilities and business perspective into account. The second Lean UX methodology, agile development is an iterative and incremental approach that enables teams to immediately respond to customer feedback, collaborate in teams, and with the customer to stimulate continuous feedback and collaboration. The third Lean UX methodology, Lean Start-up refers to the build-measure-learn loop for faster work completion. This is done through the development of a minimum viable product (MVP) to get customer feedback as soon as possible through rapid prototyping. User stories and requirement engineering previously mentioned by Forbrig and Herczeg (2015) form the basis for MVP formation argue Elberzhager et al. (2017). In this context, HCD principle can be applied to run parallel or subsequent processes for multiple MVPs with different teams.

4.8 BizDevOps and continuous BPM improvement

4.8.1 BizDevOps product and process testing

Another important aspect in BizDevOps, argue Ravichandran et al. (2016) is to make testing an accelerator rather than an obstacle for rapid application delivery with the highest level of quality. The authors propose an *Agile testing trifecta* consisting of three crucial components:

- *Test automation* methods to create requirement-driven and customer-centric test cases
- *Data management* to generate on-demand synthetic test data and drive quality improvement
- *Test constraint removal* through service virtualization of every environment that needs to be accessed.

Service virtualization in this context removes constraints by simulating constrained components in any environment and providing low-cost available models 24/7. As a result, a large part of the testing can ‘shift left’, meaning that it can be moved earlier in the software development lifecycle. *Shift-left testing* enables parallel development, testing, and validation of the software which leads to earlier defect resolution (avoidance perspective`) and faster time-to-market (Ravichandran et al., 2016).

Satyal et al. (2019) expand further on testing as a continuous improvement mechanism in DevOps by developing a two-phased AB-BPM methodology (Figure 4.10) consisting of A/B testing and simulation. A/B testing is a statistical technique used in software deployment. Two versions of a deployed product (MVPs e.g.) are compared by observing the responses of users to version A and version B to determine which one performs better. A/B testing is an effective way to continuously monitor customer feedback, to identify value-adding features, and build up resilience towards rollbacks. Satyal et al. (2019) apply the same principle also for continuous and incremental business process improvement. A simulation technique is used to extract decision probabilities and metrics from the existing process version. Based on this knowledge a new version of the process is generated and operated in parallel to the old version. A/B testing and the routing algorithm, in this case, are used for ultimate convergence towards the best performing version.

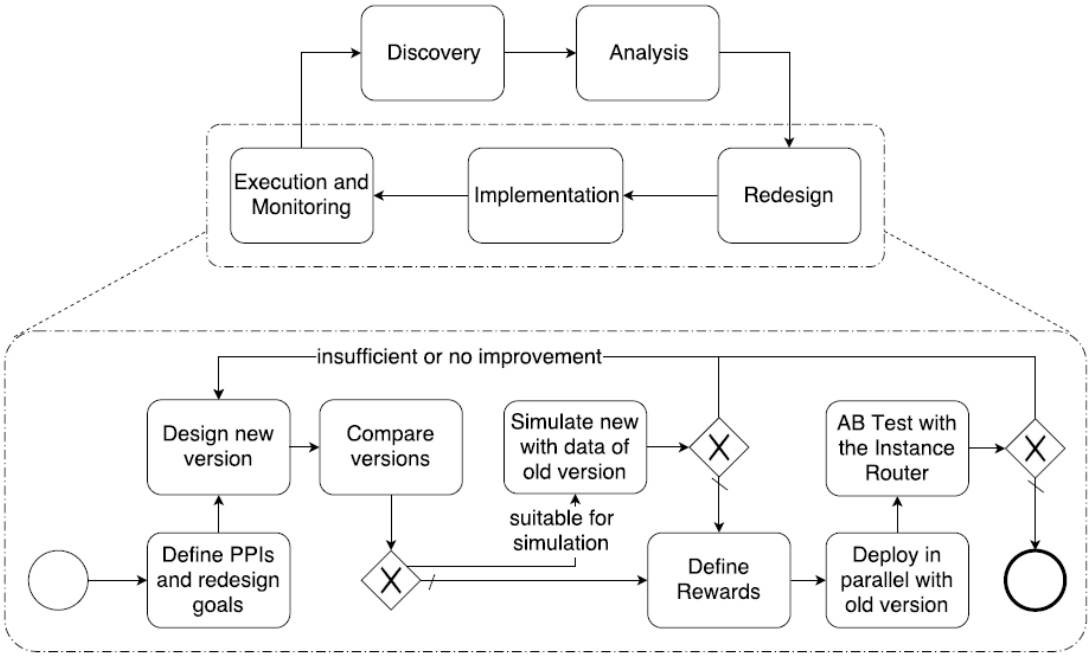

Figure 4.10 – AB-BPM methodology for business process improvement, adapted from Satyal et al. (2019, p. 287)

4.8.2 BizDevOps process modeling

Chasioti (2019) provides a high-level BizDevOps process model (Figure 4.11) that represents a software delivery process in a DevOps environment. The process model aims to address multiple stakeholder alignment and enhance user-centricity. The BizDevOps process model maps out all involved stakeholders and is split into three stages. Each of the three stages consists of several subtasks (understand, explore, prototype, etc.) and are linked in iterative cycles and First, in the *explore & identify* stage new product ideas and requirements are specified with the business unit playing the dominant role. This is also the first point where the alignment of DevOps and Business goals takes place. Second, in the *develop & operate* stage, the actual software development process takes place where the business unit plays a supportive role in development and operations. In the final stage, *validating business value*, the main goal is to evaluate the whole product development effort through metrics and outcomes.

Chasioti (2019) furthermore, addresses several limitations to the process model. For example, the process model is rather a description of activities in the software delivery cycle. The author indicates a need for designing a systematic, formal, and action-based BizDevOps Method which this thesis paper aims to do. Also. the process model only went through formative valuation with a single company (Accenture), which may question the usefulness and the general applicability of the model. Additionally, while some continuous improvement mechanisms such as performance measurement have been mentioned, the extensive elaboration on for example which approach and KPI metrics to use has not been provided. Nevertheless, the BizDevOps process model provides a solid structure and outline of activities, aspects, and stakeholders which this thesis paper will expand on to design and validate a comprehensive BizDevOps Method.

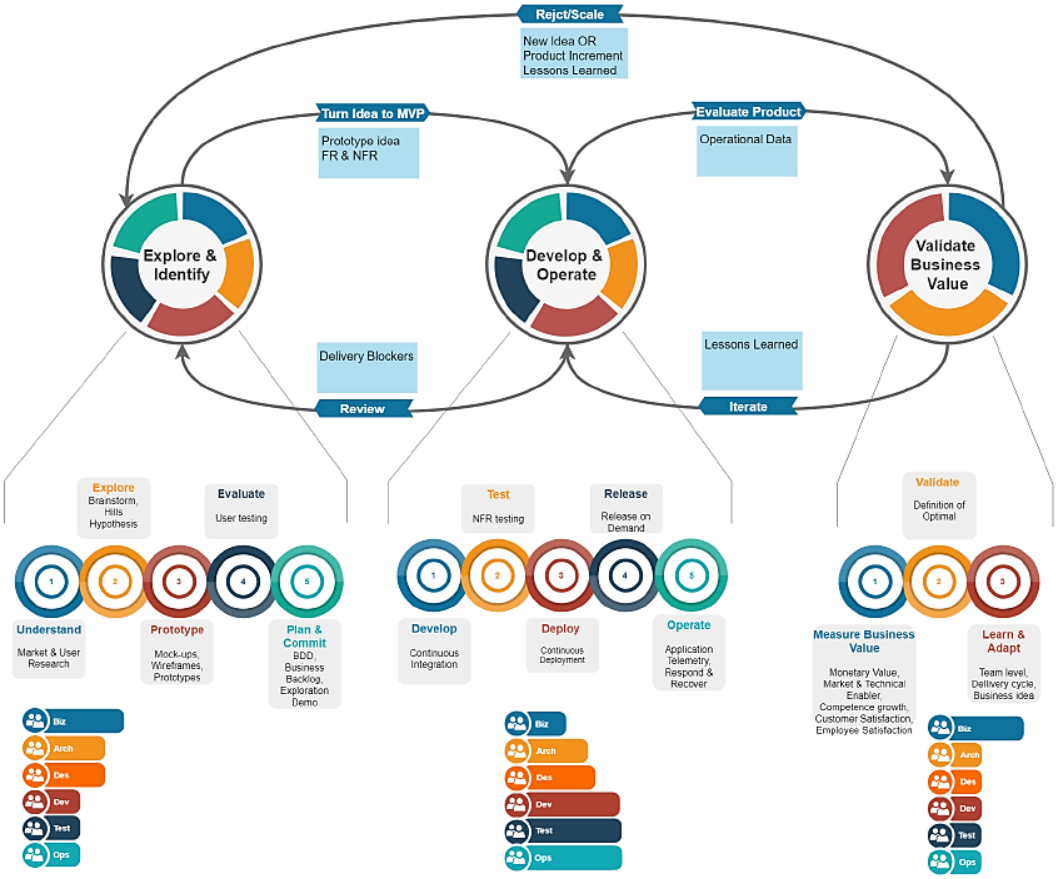

Figure 4.11 - BizDevOps process model for Scrum, adapted from Chasioti (2019, p. 62)

4.9 Capturing value with BizDevOps metrics

In the context of software development, key performance indicators (KPIs) are an important aspect regardless of the development methodology followed by an IT organization. KPIs help to control and improve the quality of the software development process and delivery. While the reviewed academic literature recognizes the need for a holistic set of measures to capture business value and quality, there is no established consensus on a universal set of agile and DevOps metrics focused on business value. According to Forsgren et al. (2018), the essence of adapting to the current dynamic environment in software development is measuring capabilities instead of maturity. They make a distinction between maturity and capability models, thereby arguing that maturity models only help organizations to arrive at a certain stage. They furthermore argue that maturity models simply measure technical proficiency or tooling which as a result makes these measures vanity metrics that do not tell anything about their impact on the business. It is essentially the capability-focused model that follows the continuous improvement paradigm and is focused on outcomes that enable leaders to set improvement goals. Capability models are multidimensional and dynamic. They enable different parts of the organization to take a customized approach, focus on improvement and capabilities, and continue to evolve with the changing environment to stay competitive.

4.9.1 DevOps and Agile unit metrics

In the work by Kupianen et al. (2015) several most occurring metrics in agile and lean software development are mentioned: *velocity*, *effort estimate*, *defect count*, and *customer satisfaction*. This

according to the authors implies that agile teams see value in planning (effort estimate), progress tracking (velocity), pre-release quality (defect count), and post-release quality (customer-satisfaction). The result of these studies brings forth two commonly reported quality-related metrics. Defect count and customer satisfaction are not directly recommended by Agile or Lean methods. Nevertheless, the two metrics are perceived as important due to their potential to compensate Agile method shortcomings in having direct quality metrics, argue the authors. Additionally, Kupianen et al. (2015) mention three metrics to measure business value on the whole level of organization: Return on Investment (ROI), Net Present Value (NPV) and Internal Rate of Return (IRR). For an overview of high-influence agile metrics please refer to Table 4.3 and Figure 4.12.

Table 4.3
Categorization of Agile metrics

ENTITIES	INTERNAL	EXTERNAL
	Products	
Products	Running tested features build status	Customer satisfaction, progress as working code
Test plans	Number of test cases	…
Code	Technical debt categories, technical debt in effort, violation of static code analysis, task's expected end date, task done, effort estimate, story complete percentage	…
Features	…	Business value delivered (ROI, NPV, IRR)
Requirements	Requirement's cost types, percentage of stories prepared for sprints	…
Defects	…	Defect trend indicator, predicted number of defects
	Processes	
Testing	Defect count, test success rate, test failure rate, defects deferred, test coverage, test growth ratio	Number of bounce backs, fault slips
Implementation	Velocity, number of unites tested, completed web pages, cost performance index, schedule performance index, planned velocity, common tempo time, check-ins per day, fix time of failed build, velocity of elaborating features	Story flow percentage
Whole	Development cycle	Cycle time, lead time, processing time, queue time, maintenance effort, work in progress, variance in handover, throughput, queue, implemented vs wasted requirements
	Resources	
Team	…	Team effectiveness
Customer	Revenue per customer	…

Note: data for agile metrics adjusted from Kupianen et al. (2015, p. 151)

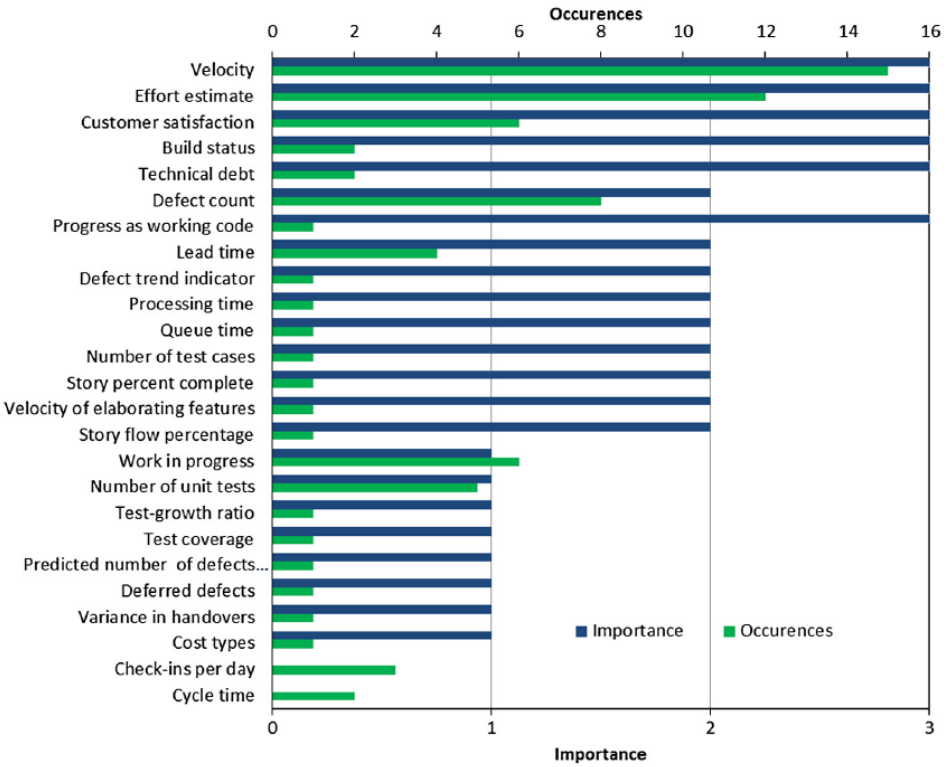

Figure 4.12 – High influence agile metrics based on the number of occurrences and perceived importance, adjusted from Kupianen et al. (2015, p. 155)

Regarding velocity, Forsgren et al. (2018) explain that in many schools of Agile, problems are broken down into stories to which developers assign several points representing the relative effort expected to complete them. The total number of points signed off by the client at the end of an iteration is recorded as the team's velocity. Forsgren (2018) in this context agrees with Kupianen in the sense that velocity can be used as a capacity planning tool to estimate the time needed to complete the work and to track progress. However, Forsgren (2018) warns against the practice of using velocity as a productivity metric or a way to compare teams. First, teams usually have a substantially different context which makes their velocities incommensurable. Second, teams tend to inflate their velocity estimates and narrow down their focus on story completion. This can jeopardize the collaboration with other teams and may even create internal competition ending up in a silo mentality. Utilization is another problematic productivity measure identified by Forsgren (2018). While it may be good up to a certain point, after a certain level there is the no longer spare capacity (slack) to absorb unplanned work, changes to the plan, or improvements. This causes longer lead times to complete work.

Forsgren (2018) furthermore proposes 4 essential DevOps metrics for delivering business value:

- *Lead time of change* – the evaluation of lead time is the key metric in Lean theory. In the software development context it is defined as the time it takes to go from code commitment to the code successfully running in production. Forsgren (2018) argues that shorter lead times are better since they enable faster feedback, more rapid and confident adjustments on what is being built

- *Deployment frequency* – is a proxy to batch size in Lean theory and it measures the frequency of software deployment (changes that get deployed) to production. Usually, a release consists of multiple version control commits, unless the IT organization has achieved continuous deployment

with a single-piece flow. It is argued that reduced batch sizes decrease cycle time, variability in flow, reduces risk, and overhead. At the same time feedback is accelerated, efficiency is improved and motivation is increased.

- *Mean time to restore (MTTR)* – refers to the average time it takes to restore service and as such it serves to investigate whether the teams who have improved their performance in terms of lead time have done so at the expense of system stability. Therefore, MTTR is seen as an essential quality metric in scaling DevOps for business value generation.

- *Change fail rate* – this metric looks at what percentage of changes made in the system fails. In the context of Lean, this is the same as completeness and accuracy of product delivery and process, hence making change fail rate another key quality metric.

Elliot (2014) argues that communicating business value with DevOps is critical for enterprise agility and growth. In a research survey from November 2014 (Table 4.4), Elliot outlined the following industry best-practice BizDevOps specific KPI metrics derived from over 20 Fortune 1000 companies:

Table 4.4
Fortune 1000 Business DevOps KPI metrics

Productivity related metrics:	○ Speed ○ Velocity ○ Impact analysis ○ Build and test automation ○ Configuration automation ○ Time to market
Quality related metrics:	○ Availability ○ Requirements analysis ○ Business stakeholder ○ Involvement and support ○ Metrics that help to identify issues earlier through continuous testing and integration
Operating expense related metrics:	○ Accost optimization ○ Cost modeling ○ Metrics that encourage fail fast and fail cheap ○ Allocation of IT
Capital expense related metrics:	○ Utilization ○ Cloud-based systems and convergence

Note: data for business DevOps KPI metrics adjusted from Elliot (2014, p.12)

4.9.2 Frameworks for (Biz)DevOps KPI measuring

Ravichandran et al. (2016) note that DevOps metrics should be characterized by being obtainable, reviewable, incorruptible, and actionable (support decision-making improvement). Furthermore, the

author provides a four-dimensional framework (Figure 4.13) to measure DevOps metrics in a business context. The framework takes people, process, and technology into account and reflects the metrics through the following four dimensions; 1) *culture, collaboration & sharing,* 2) *efficiency & effectiveness,* 3) *quality & velocity, and* 4) *customer & business value*.

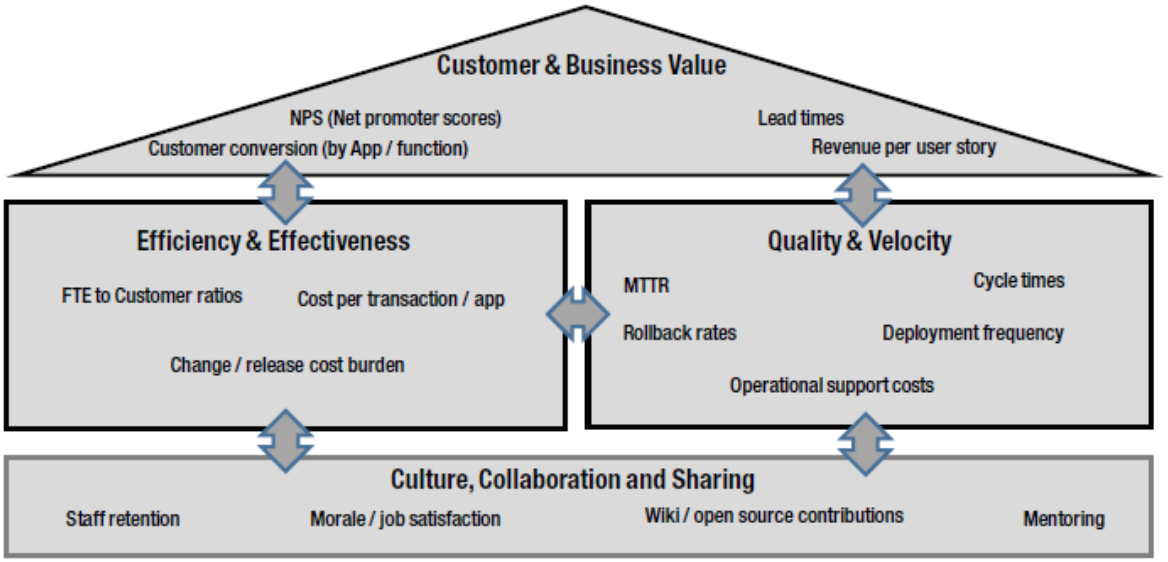

Figure 4.13 - DevOps metrics dimensions, adapted from Ravichandran et al. (2016, p. 44)

Metrics in the culture category are essential because they provide a continuous indicator of acceptance/resistance to DevOps in the organization. Some relevant metrics given in this dimension are staff retention and job satisfaction among others. Efficiency and effectiveness dimension is channelled in a customer-centric fashion so that inefficiencies associated with acquiring, preparing, and maintaining the infrastructure for development, testing, and production can be exposed and tackled appropriately (Ravichandran et al., 2016). An example of a metric in this dimension includes a full-time equivalent to the customer (FTE). Quality and velocity measure data points connected to service delivery with the focus on establishing quality from the start of development to for example reduce roll-backs and code defects as early as possible. Several useful metrics such as cycle time and MTTR among others are mentioned in this dimension. Finally, customer and business value as the roof of the framework are externally oriented metrics that help measure how DevOps supports business goals such as increasing customer loyalty and time-to-market. Just like in the works of Kupianen et al. (2015) and Forsgren et al. (2018), this framework stresses the critical importance of lead time as it provides DevOps teams with an analogous metric to determine how well the need for rapid delivery of high-quality software service is being met (Ravichandran et al., 2016). Another interesting metrics is Net Promoter Score (NPS) which measures customer loyalty that also determines how quickly services can be delivered.

Having identified metrics across all four dimensions, Ravichandran et al. (2016) propose business impact mapping to gain insights into what processes and tools are needed to address capability gaps and meet targets. This is a simple and effective approach to determine which interrelated DevOps processes, metrics, targets, and initiatives are needed to support business goals. Figure 4.14 illustrates a case where an organization aims to achieve an NPS through several metrics, targets, and initiatives.

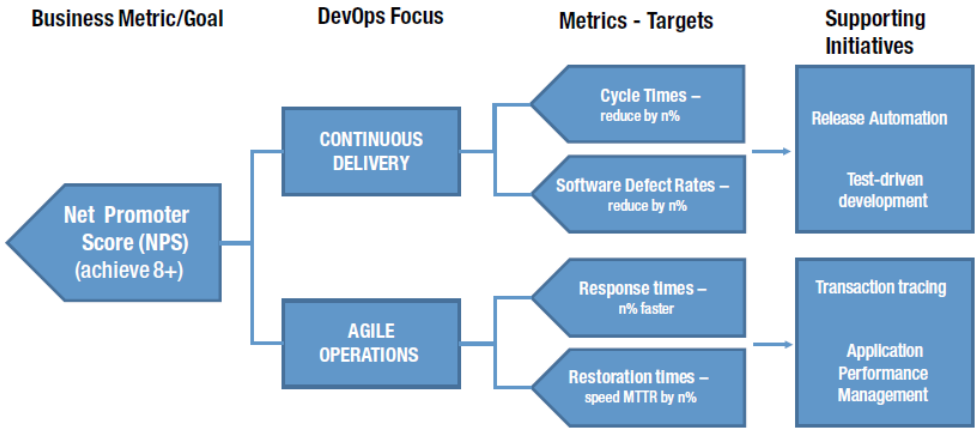

Figure 4.14 – An example of DevOps business impact mapping, adapted from Ravichandran et al. (2016, p. 46)

Korpivaara (2020) arguably provides the most complete and extensive framework (Figure 4.15) for selecting and measuring KPI metrics in agile software development. The framework integrates a stakeholder-driven value chain perspective by Porter and a Balance Score Card model (BSC) which is fairly similar to the business impact mapping, proposed by Ravichandran et al. (2016). Furthermore, Korpivaara's framework takes a multi-dimensional and multi-layer approach and makes a distinction between external and internal metrics, which was previously seen by Kupianen et al. (2015) and Ravichandran et al. (2016). This framework further extends previous models and insights in this systematic literature review by providing prioritization categories. Primary performance objectives measure the impact on the end-beneficiaries in the value chain, which in this case are customers and owners. This category indicates the ultimate value and therefore includes the most important objectives. However, the metrics within this category are lagging and indicate historical rather than future performance. Secondary performance objectives measure how efficiently activities and operations are performed, which corresponds to productivity and quality-related metrics previously mentioned by Elliot (2014), Kupianen et al. (2015), Ravichandran et al. (2016) and Forsgren (2018). Similarly to Ravichandran et al. (2016), Korpivaara (2020) takes culture, collaboration, and learning as key performance enablers, focused on capabilities and resources. These indicators are leading as they indicate future performance better than the performance objectives.

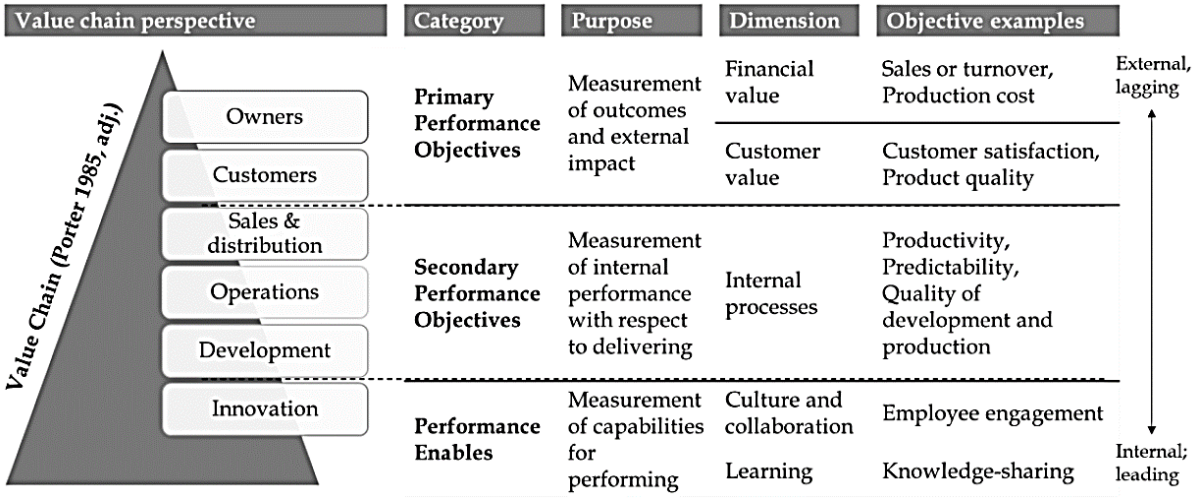

Figure 4.15 – Framework for measuring performance in Agile software development organizations, adapted from Korpivaara (2020, p.62)

4.10 Chapter conclusion & discussion

Based on the conducted multivocal literature review, it can be concluded that next to a limited amount of available empirical research, there has been barely any suggestion of a holistic and solution-based artifact for the alignment of DevOps with business goals. Furthermore, no elaborated strategy for addressing the Business and IT gap have been covered. The literature rather provides loosely related concepts and practices which nevertheless have value in answering knowledge questions KQ1 and KQ3. In addition, the literature provides a good knowledge foundation and a basis for comparison with empirical findings which allows this research to bridge theory and practice in developing a strong BizDevOps artifact.

Section 4.3 provided a general understanding and relevance of BizDevOps as a concept. Sections 4.4 and 4.5 expanded further on the core DevOps practice of having cross-functional teams, thereby examining how to more actively involve business stakeholders in the process. Next to the usual PO position, several other roles of business stakeholders have been suggested such as business designer, business ambassador, and business expert (on-site-customer). Having elaborated on the relevance and contribution of business stakeholders, the literature provided valuable insights to answer KQ1. BizDevOps thus implies a cross-functional team whereby the business stakeholders are active participants throughout the whole process. They are versatile individuals capable of translating user needs into business requirements, and continuously updating and managing the requirement backlog. Biz in DevOps are business value maximizers who are capable of prioritizing tasks and resources to shorten and simplify feedback cycles that result in optimal service delivery. Section 4.6, 4.7 and 4.8 provided additional insights on how to optimize information flow within a BizDevOps software development context. BizDevOps teams should rely on additional continuous engineering practices such as continuous planning, requirements engineering, and human-centered design among others. As a result, IT organizations can develop high responsiveness to changing business needs in order to adjust their workflow and deliver better end-user experience.

Finally, literature from section 4.9 addressed KQ3 which refers to the issue of capturing business value with metrics. While no universal metric package has been identified, the obvious pattern in the literature indicates a symbiosis of result-oriented metrics across cultural, performance, and customer domains to measure the effectiveness of DevOps in capturing business value. Several highly influential performance metrics such as velocity and MTTR, as well as business-specific metrics such as revenue per customer, NPV, and ROI have been identified. Additionally, a number of models for measuring KPIs towards business impact have been presented. Therefore, these insights give a good foundation and a non-exhaustive list of metrics that will further be extended and refined. Nevertheless, considering the limited amount of available literature on business-centric DevOps KPI metrics and no common structure, additional empirical research is needed to provide a more complete set of insights needed for answering all knowledge questions. Empirical research should produce practical, more detailed (best)practices, processes, metrics, and challenges in aligning DevOps with business goals and enhancing enterprise agility. In this way, a symbiosis of theoretical insights gained from the multivocal literature review and practical insights gained from the empirical study provide stronger grounds for designing a robust and comprehensive BizDevOps artifact. Practices identified from the systematic literature review can be found in Table 4.5 on the following page.

Table 4.5
Practices identified from the literature

THEMATIC CATEGORY	PRACTICES IDENTIFIED
4.3 Defining BizDevOps	• Merging and aligning business, development, and operations
4.4 BizDevOps cross-functional teams	• Active involvement of business stakeholders in cross-functional teams
4.5 The role of Biz in BizDevOps	• Low-code Shadow IT • Requirement backlog management • Business tracker, ambassador, designer • Business expert on-site-customer
4.6 Continuous in BizDevOps	• Continuous planning • Continuous innovation • Continuous improvement
4.7 Subject-oriented approach in BizDevOps	• Continuous requirements engineering • Continuous business process modeling • Human-Centered Design • Storytelling • Lean UX
4.8 BizDevOps and continuous BPM improvement	• Agile testing trifecta • Shift-left testing • A/B testing • AB-BPM • Business value validation
4.9 Capturing value with BizDevOps metrics	• Customer-centric KPI measuring • Capability orientation over maturity • Multi-dimensional KPI measuring • Business impact mapping • Stakeholder-driven value chain perspective • Balance Score Card • Priority categorization

5. QUALITATIVE STUDY - BIZDEVOPS IN PRACTICE

This chapter presents the findings from the qualitative study interviews.

5.1 Defining interview respondents

Due to the circumstances surrounding the Covid-19 pandemic, the limitations of physical contact, and pivot towards adaptation and survival in many organizations, the initial goal was set at finding 5-10 interview respondents with relevant experience. To do so, multiple channels were used to find and reach out to potential respondents. This for example included position-based targeted search on LinkedIn, leveraging own networks and networks of the respondents, tracking the authors of whitepapers, blogs, webinars on BizDevOps, etc. A special software called RocketReach was used to obtain working contact details of potential candidates and all the information was stored in an excel dataset from which respondent engagement management, cold messaging, and mailing started. For privacy purposes, this dataset will permanently be deleted after the end of this research. Additionally, to respect the respondents' privacy and research ethics, each practitioner has been assigned a unique code and a pseudonym for their organization based on the related industry sector. This approach to scouting for respondents surprisingly resulted in 13 interviews with practitioners who work at different organizations, ranging from IT services, product companies, banking, IT research, and the government among others. Regarding the profile of the respondents, they have a diverse background across the business and/or the IT side of the spectrum, with their experience level mostly ranging from 10 to 20+ years in the industry (the minimum experience rate of 5 +/- years was only encountered in the case of one respondent, IR-5). This means, that this qualitative study captured insights from both practitioners with lower working seniority as well as from practitioners in senior management positions. The following table outlines all codified respondents, their organization's pseudonyms, and positions.

Table 5.1
Overview of qualitative study respondents, their positions and organizations

CODE	ORGANIZATION PSEUDONYM	FUNCTION
IR-1	Low-code App Development Platform	IT Architect
IR-2	IT Research Institute	DevOps Research Analyst
IR-3	IT Monitoring Solutions 1	Senior Product Manager
IR-4	IT Quality Services	Scrum Master/Agile Coach
IR-5	IT Bank	Product Owner
IR-6	IT Education and Learning	Product Owner/Business Analyst
IR-7	IT Surgery Solutions	Chief Product & Technology Officer
IR-8	IT Sourcing	Scrum Master
IR-9	IT Fintech Platform	Senior Product Manager
IR-10	IT Government	Manager IT Advisory Unit
IR-11	IT Stats	Agile Coach
IR-12	IT Operation Services	Release Manager/DevOps Consultant
IR-13	IT Monitoring Solutions 2	DevOps Engineer/Activist

5.2 Qualitative study approach

Most of the respondents were reached by e-mail with a short introduction of the topic, the interview protocol, and a request to participate in the interview. All interviews were conducted online using Google Meets and lasted on average 30-40 minutes. As noted previously, the interviews were semi-structured, meaning that some deviation from the standard set of questions was allowed, depending on the flow of the conversation. The conversations were recorded with respondents' consent, using Otter.ai which leverages artificial intelligence to translate audio into *verbatim* transcripts in real-time. The software proved to be highly accurate, demanding only minimal editing, and even provided a set of identified keywords that contributed to the analysis. The interviews were analyzed, following a mixed principle of deductive and inductive coding described by Seale (2004). A deductive approach was taken to identify some themes and develop codes before the interviews. The literature and research questions were used as the basis for developing a codebook. During the analysis, a more inductive approach was taken to observe frequently mentioned words, patterns, and contexts to derive new codes. *Open coding* was applied to chunks of text to generate concepts and categories. These categories served to identify practices that are essential in aligning DevOps with business goals.

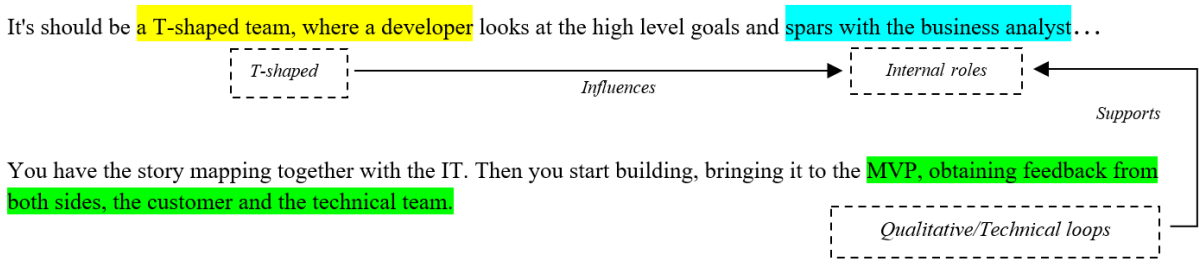

Figure 5.1 – Example of coding interview transcripts and determining relationships between the codes I

After the open coding process, axial coding was used to classify codes into broader categories and determine relations between them. These categories represent high-level processes and concepts, essential in BizDevOps. The relationships between categories were derived based on the relationships from open codes belonging to the category. Nevertheless, it should be noted that as a result of such an approach, there might be additional relations that support, hinder, or interact that have been left out. Therefore, it is assumed that the relationships are partially true and their nature is not included in the code-based conceptual map that summarizes insights from the qualitative study interviews.

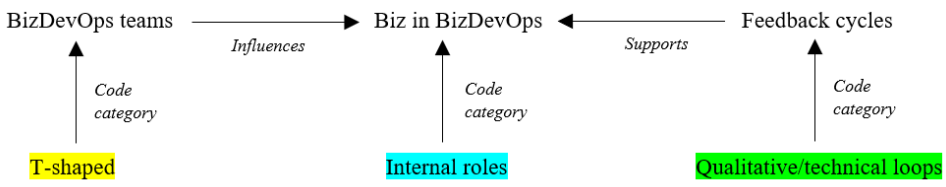

Figure 5.2 – Example of coding interview transcripts and determining relationships between the codes II

The outcome of the analysis resulted in 33 open codes, which were grouped in 9 broader categories. These results and relationships have been summarized in a conceptual map, found in Figure 5.3 on the following page.

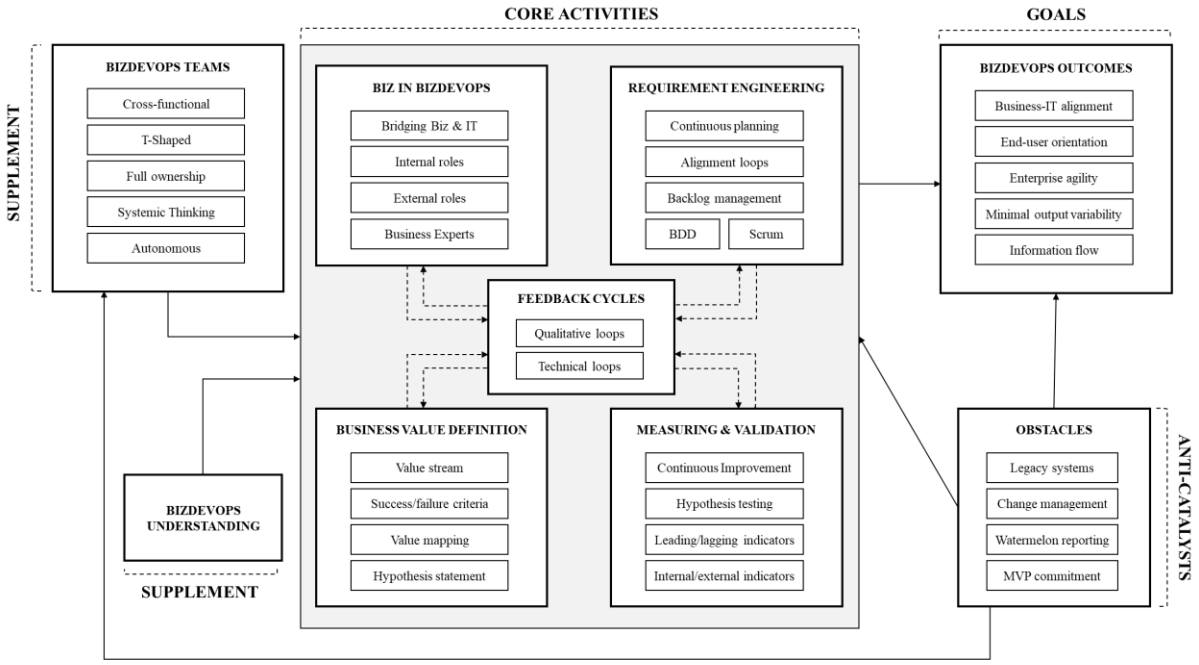

Figure 5.3 – Conceptual map of the qualitative interview results

According to the conceptual map, BizDevOps understanding and BizDevOps teams have been identified as supplements (inputs) that provide the base and support for the core activities. Therefore their arrows point directly to the core grey area, meaning that they influence all blocks in this zone. The broader categories in the grey area, Biz in BizDevOps, business value definition, requirement engineering, measuring & validation, and feedback cycles have been identified as the core for achieving the desired outcomes (arrow pointing from the grey area, towards the BizDevOps outcomes). All blocks in the core are connected through feedback cycles which are represented as a crossroad that connects the corridors for providing and receiving input. These corridors are represented by eight two-way dashed arrows, pointing from each block towards the feedback cycle and pointing away from the feedback cycle, towards each of the four core blocks. Feedback cycles in this context serve as a critical alignment loop between stakeholders and processes that have an impact on the achievement of desired outcomes, mapped in the BizDevOps outcomes block. Finally, the last category mapped out as 'obstacles' can be explained as an anti-catalyst. Its arrow pointing for example towards the 'BizDevOps teams' means that obstacles negatively impact certain stakeholders or activities connected to this block. Therefore, obstacles represent a factor that possibly hinders the achievement of desirable outcomes (arrow pointing from the obstacles, towards the BizDevOps outcomes).

5.3 Common understanding of BizDevOps outcomes

Even though the BizDevOps outcomes block is positioned on the far right in the figure, it is important to start with it first, together with BizDevOps understanding, since these two blocks provide the 'WHY' of this research project and success indicators of BizDevOps. The practitioners' view surrounding BizDevOps is very much in line with the literature results covered in section 4.3. According to a strong majority of respondents, BizDevOps is commonly understood as an approach or rather a mindset that bridges business, development, and operations. BizDevOps serves as a Business-IT alignment mechanism that aims to unify the three fields around a common goal and bring down the last silos between business and IT. IR-5, IR-9, IR-4 note that the implementation of BizDevOps should result in

the enhancement of enterprise agility, development with the end-user in mind, continuous feedback, information flow, and minimized product output variability. IR-9 notes that BizDevOps if properly implemented allows the team to make a better balance between the level of detail (quality component) and time constraints. IR-5 for example had the following to say on the matter of BizDevOps:

> *"In BizDevOps teams we make sure that the alignment and communication between business and DevOps is very close and that is why we put them together. They continuously communicate about the functionality of the system and user-friendliness, they work together to implement changes and improvements. So it is about having a continuous feedback loop."*

5.4 BizDevOps teams

The common vision of the interview respondents regarding the characteristics of the BizDevOps team is that they are cross-functional, a concept that appeared earlier in the literature findings section 4.4. Moreover, multiple respondents (IR-4, IR-6, IR-11) brought up the concept of autonomous T-shaped teams. This implies that each member of the BizDevOps team next to having a depth skill, also possesses the breadth and understanding across all other disciplines in the team. For example, an Ops in a T-shaped team next to being specialized in functional testing also has the basic knowledge about development and can fulfill the simplest task that a developer does (IR-4). Moreover, IR-4, IR-9, IR-1, IR-3, IR-10 brought up full ownership, systemic thinking, and development with the end-in mind as integral components of team mentality in BizDevOps.

> *"You build it, you run it. BizDevOps teams have full ownership of the process and a mindset that focuses on creating the software with the end in mind. It is about thinking of the process not project." (IR-4 and IR-9) (IR-1) (IR-3)*

Several respondents thus claim that BizDevOps teams should own and be able to envisage the whole process from defining the goals and requirements, development, delivery pipelines (CI/CD), conflict management, logging, monitoring in production, and reporting of results.

5.5 Biz in BizDevOps

This section introduces the findings concerning the role of the business unit within BizDevOps

5.5.1 Bridging Biz and IT

The role of the business unit 'Biz' within the BizDevOps was generally understood as the bridge between the client (the end-user), and development (Dev) and operations (Ops). The practitioners show a consensus with the literature findings that Biz as a bridging unit within BizDevOps, it is imperative that the business unit is continuously involved in the whole process. For example, the critical contribution of the business unit in the beginning stages includes the value definition and refinement of requirements for the IT unit. In the later stages of the process, Biz is an essential line that provides insights on the system usability, suggestions for a better product-market fit, and validates the value delivered. Moreover, during the entire duration of the development process, Biz manages a variety of stakeholders from the client, Dev, Ops, senior management, and external vendors for example. Several specific roles; both internal and external corresponding to the business unit have been identified by the interview respondents.

5.5.2 Internal Biz roles

The most frequently mentioned Biz role was the Product Owner who serves as the catalyst in connecting the end-user with IT and manages a number of stakeholders during the process. The Product Owner defines the product backlog, breaks down high-level business requirements, and determines important functionalities that should be featured in the system. To do so, the Product Owner participates in product management meetings and cooperates with the Product Manager who owns the business vision, the roadmap, product/value delivery and is responsible for keeping the process geared towards the end-user. Business Analyst is another prominent role brought up by five respondents. Business Analyst works hand-in-hand with the Product Owner and uses his/her knowledge of the market to scope the size of the project, define use-cases, translate and refine the business requirements into stories for the IT unit. These descriptions of responsibilities of the Product Owner, Business Analyst, and Product Manager align with the roles of Business Tracker and Business Ambassador covered by the literature in section 4.5. Moreover, several respondents provided extended insights into the role of the Product Owner. Next to the aforementioned responsibilities, the Product Owner is an active participant in iteration planning, team retrospectives, product demos, and implements story improvements. Finally, IR-4 and IR-11 acknowledged the role of the Agile Coach/Scrum Master who serves as the facilitator of high-performing BizDevOps teams. This includes the promotion of agile practices, conflict resolution, and aligning the teams around the same goal.

5.5.3 External Biz roles

Next to internal Biz roles, the majority of interview respondents stated that their organizations extensively rely on parties that are not necessarily a part of the BizDevOps teams but are equally important for maximizing value delivery. IR-5 from the IT Bank for example indicated that the users of their systems are a major communicator from which their BizDevOps team gets a lot of input for improvements. This notion was also supported by the respondents from product companies, IT Monitoring Solutions, and IT Education and Learning. Both companies organize customer panels and utilize customer advisory reports to capture the end-user perspective in the adoption of new features for the system. This practice can be related to the concept of *Customer-on-site,* covered in section 4.5 of the literature review. Additionally, IR-13 noted that Sales Engineers are also a source of valuable insights as they are the first line in the field and know which features sell well. IR-1 from another software company, Low-code App Development Platform mentioned the use of Customer Success Managers as another prominent way to gather key end-user input and maximize business value.

5.5.4 Business Experts

Before moving on to the next section, it is essential to address the role of a Business Expert that was covered both in section 4.5 by literature and by interview respondents. IR-6 from IT Education and Learning saw this role as follows: *"Business Experts can play a big role in requirement and feature specification because they have the experience on how the system functions."* Based on the qualitative study findings, the role and nature of a Business Expert vary from organization to organization. For example, IR-11 from IT Stats noted that they utilize Business Experts as key users internally and they can directly be a part of the BizDevOps team or can be outside of the team, but participate in different meetings to assist with refinements and iterations. IR-4 from IT Quality Services noted the following: "*Key users are the ears and eyes of the organization that communicate how the system is functioning*." Sales Engineers and Customer Success Manager brought up by IR-13 and IR-1 are another example of

an internal Business Expert. On the other hand, an example of an external Business Expert was demonstrated by the *Customer-on-site* role that was seen in the case of IT Monitoring Solutions and IT Education and Learning.

The variation in the role of a Business Expert and correspondent's contrasting opinions in where this role should be positioned can mostly be explained by the business model and the sector of the organization. It has been noted that product companies such as IT Monitoring Solutions, IT Education, and Learning prefer to have this role fulfilled more externally by Sales Engineers, Customer Success Managers, and even the customer. On the other hand, IT service organizations and organizations with IT departments such as IT Quality Services and IT Bank have shown preference to have the role of Business Expert as a key-user within the team or as a separate entity.

5.6 Business value definition

The following section covers the gathered insights concerning business value definition and scoping in the BizDevOps setting.

5.6.1 Value stream

According to IR-2 from the IT Research Institute, having a rich and holistic understanding of the business value stream is an essential component for accurately defining value in BizDevOps. As a result, value stream maps should take the following four aspects into account: people, process, technology, and data as the roof. The people aspect in this situation refers to the stakeholders, the effort to align their goals, and to enhance cooperation. Process refers to the activities and steps undertaken to deliver high-quality software, the effort to optimize and align these with the business strategy. Technology implies the capabilities and tools such as CI/CD to support development and challenge tackling. At last, the data aspect represents the information flow and the capability of the BizDevOps teams to observe what is happening with the system and processes so that necessary improvement mechanisms can be triggered.

5.6.2 Defining success/failure criteria

Another critical element in business value definition according to a number of respondents is the definition of success and failure criteria. IR-3 from IT Monitoring Solutions stated the following regarding this matter:

> *"You need to understand how does success but also failure look like. If you know what the business goal is, you can translate it into something that you can measure and what you can measure, you can evaluate and adjust based on the feedback that was received."*

In other words, clearly defined success/failure criteria, enable agile organizations to establish continuous improvement mechanisms, capture value, and performance insights through a set of KPI metrics to validate work and make necessary improvements.

5.6.3 Value mapping and hypothesis statements

When asked about techniques to scope and define business value, high-level product road mapping and business impact mapping were the two predominant practices mentioned by the practitioners. High-

level agile product roadmaps, for example, enable the stakeholders to effectively communicate short-term and long-term strategic goals. According to IR-11 from IT Stats, high-level product roadmaps are defined on the level of epics and are focused on the outcome. Lower-level roadmaps are reserved for the product backlog on the level of the BizDevOps team. Moreover, IR-13 from IT Monitoring Solutions gave an example similar to the literature (Figure 4.14) on how business impact mapping can be used to define a goal. For example, they once defined the success criteria for product usability at 50% of the customer base. This means that anything above this level of usability indicates success. The process of business value definition often includes a hypothesis statement which is later on tested for value validation, as noted by IR-13 and IR-6 among others.

5.7 Requirement engineering in BizDevOps

The following part presents practical insights concerning the translation of end-user requirements to technical specifications, their refinements, planning, and alignment with business goals.

5.7.1 Continuous planning and alignment loops

Once the high-level business goals have been defined, validated, and aligned with the senior management, it is up to the Product Owner to breakdown business requirements into lower, feature-level, user stories and use cases. IR-11 and IR-6 note that this usually happens in close cooperation with the Business Analyst, Lead Dev, and Business Expert. IR-3 specifically advocated that the requirement specification should be approached with an iterative thought, which is something that frequently appeared in the literature. Several interview respondents expanded further on this notion by giving an example of their iterative approach and continuous planning through Agile PI Events. IR-10 for example stated that PI events are a good way to align the vision and goals of the business across the stakeholders as well as to plan which stories and iterations the BizDevOps teams should commit to delivering. Moreover, two interviewed Product Owners highly emphasized the presence of the alignment loop during this process. Requirement specification, story refinements, and scoping should continuously reflect high-level business goals, note IR-4, and IR-6. Another noticed practice indicative of an iterative approach by the respondents is the concept of prototyping. IR-12 and IR-13 saw prototypes such as mock-ups for example as an effective way to obtain rapid feedback from the end-user, rather than finishing up the whole system, only to find out that it does not meet the business requirements. What is more, all respondents admitted to using MVPs, which is another indication of an iterative approach that was also seen in the literature.

5.7.2 Backlog definition and management

Naturally, not all requirements and stories can have the same level of priority in the backlog. It is therefore up to the Product Owner to coordinate with the rest of the BizDevOps team which features should the team commit to first. A frequently mentioned way of assessing a feature vis-à-vis the high-level business goals is the Weighted Shortest Job First model (WSJF). For example, IR-6 described WSJF as a means to score a business value of a feature and prioritize the backlog. The respondent mentioned three major components of the model. The first component is the *user's business value* which can be displayed as monetary value, customer satisfaction, positive or negative impact. The second component is *time criticality* which looks at what milestones and deadlines need to be met. The final component is the *risk and opportunity* which examines the potential risk that the feature might bring but also the business impact that the new feature might cause. For instance, a new business opportunity

or a new business model. WSJF, therefore, represents one of the previously mentioned alignment mechanisms (alignment loops) that are important for a strategically-driven backlog and the prevention of deviation from the high-level business goals. Finally, when it comes to a program for product backlog management and requirement specification, JIRA is a popular tool mentioned by IR-7, IR-3, and IR-1 among others.

5.7.3 BDD and use cases

Another important alignment mechanism for BizDevOps requirement engineering brought up by the interviewed Agile Coaches, IR-4 and IR-11 is Behavior-Driven Development (BDD). It seemingly draws similarities to the subject-oriented approach and HCD covered in section 4.7 of the literature review. Namely, BDD was seen as a way for the Biz to specify end-user requirements by clearly describing the user, his/her behavior, use cases of the system, and outcomes. IR-4 for example noted that feature files following a BDD guideline are written down with a special domain-specific syntax that outlines different scenarios based on what the user does in the application. Once the behavior and outcomes are described, acceptance criteria on the feature-level can be set. Moreover, IR-13 argued that this process does not only include functional requirements of the software but also non-functional requirements such as scalability and the business impact such as user engagement and experience. Such domain-specific language that enables the business unit to express requirements also draws upon similar concepts encountered in the BizDevOps platform, covered in section 4.5 of the literature review.

5.7.4 Scrum XP

Of 13 interviewed practitioners, all of them recognized Scrum as the method for determining software builds and iterations. The process described by the respondents shows a similar pattern of activities to the HCD approach that was addressed in section 4.7 of the literature. Several key events noted by the respondents play an important role in supporting the alignment loop. The teams generally start with a sprint planning session where they determine which stories or product increments are to be built in the following 2-3 weeks. In the implementation phase, the BizDevOps team works on developing the feature and constantly communicating and aligning work during daily stand-ups, as noted by the respondents. Agile Coach or the Scrum Master (IR-4 and IR-11) plays an integral role during the whole Scrum process, as someone who constantly supports the BizDevOps team, promotes Scrum and agile practices, and takes the effort to maximize the agility of the teams. Finally, in the review and retrospective phase, the team works on evaluating the delivered features and stories and works on the implementation of continuous learning and improvement principles.

5.8 Measuring and validation of business value in BizDevOps

The section below contains the findings on measuring and validating business value delivered by the BizDevOps teams.

5.8.1 Continuous improvement

The interview findings show that performance and value measuring seems to be the most challenging aspect for the participating organizations. While some of them have already established a solid measuring foundation, others are in the exploration phase and are looking for a more standard set of metrics to track and validate the business value. Therefore, this qualitative study was able to capture

both the best practices perspective as well as the KPI requirement perspective of the participating organizations. According to IR-4 and IR-8 for the organizations that work with BizDevOps, it is imperative to establish a strong continuous monitoring mechanism that captures realistic performance, value indicators and serves as a continuous alignment and feedback loop. IR-4 gave an interesting statement on this matter:

> *"I would say that your first feedback is not personal feedback, but is in metrics. If you have good metrics, you can see how your system is used, and you can put good monitoring on it. If you always have 1000 views per day on a certain page or at the end of a certain funnel, and all of a sudden, you have 100 or zero, you want to get notification from that, you want the big red alarm to go off because something is probably wrong. So it's up to you to build in, those checks and balances."*

For example, IR-3 from IT Monitoring Solutions advocated for real-time tracking of KPI metrics. IR-8 further supports this argument by stating that a realistic and updated picture of performance and value indicators reduces the risk of watermelon reporting, which is a practice of reporting strong (green on the surface) progress, whereas in reality there are major issues with delivery (red on the inside). IR-3 also made a strong statement regarding the prioritization of business value, which is essential in capturing KPI metrics: *"Don't try to solve five-dollar problems. Fix the million-dollar problems.*"

5.8.2 Validation through hypothesis testing

Referring back to the practice of hypothesis statement covered earlier in section 5.6, IR-7 and IR-8 argued that an effective way to validate the business value delivered is to test the hypothesis that was stated during the business value definition phase. As they note, it is essential to identify events, features, and a proper set of indicators that are possibly correlated to increased customer value and test the significance of the relationship. Several respondents, in this case, talked about converting team productivity and predictability to usability metrics and eventually to financial metrics such as ROI and revenue. IR-4 and IR-7 admitted that this is one of the biggest challenges for each organization and recognized that cases, where KPI metrics are intensively tested for correlation, are seldom. Nevertheless, the majority of respondents agree that using KPI metrics independently without taking correlations into account is likely to turn those into vanity metrics which prevents organizations to focus on what matters the most, deliver the value that was initially asked by the end-user. Furthermore, IR-6 made an important remark, regarding value tracking and validation. As a long-time Business Analyst and Product Owner, IR-6 noted that validation of value is not something that happens overnight. One should strive to track and validate value over a longer period to not just confirm but also exploit new possibilities for additional value creation.

5.8.3 Leading and lagging indicators

Before we can talk about the indicators on a unit level, it is important to make two important distinctions based on the insights provided by the practitioners. A first distinction that was noted in the practices of different participating organizations is the distinction formalized by IR-2 as leading and lagging indicators, something that was also addressed in section 4.9 of the literature review (see Figure 4.15). Leading indicators are shown to play an important role in the value scoping and definition phase as they provide inputs into what actions are necessary to achieve certain goals. More specifically, business impact mapping and product road mapping practices are events that are highly dependent on the formulation of leading indicators. Therefore, leading indicators serve as a benchmark to examine

whether the BizDevOps team is meeting business objectives, notes IR-2. Lagging indicators are therefore measures that track current output and performance. They are thus used in conjunction with leading indicators for validating whether the BizDevOps team is delivering on business goals. The example previously given by IR-13 regarding the goal of having a 50% customer base using a feature is a representation of a leading indicator (success/failure criteria). Naturally, a lagging indicator, in this case, is the actual % of the customer base using the new feature.

5.8.4 Internal/external indicators

The second distinction that could be made based on the insights from the practitioners is internal and external indicators.

Internal metrics

In this sense, the internal indicators are referring to cultural, team aspects, internal performance, processes, and capabilities of the BizDevOps teams. Internal indicators have shown to be industry agnostic, meaning that they have been frequently brought up by respondents from various industries. For example, IR-1 outlined several culture/team metrics such as internal NPS and employee satisfaction that are also in line with the literature findings. Interestingly, IR-1 expanded on this notion by suggesting qualitative surveys to measure the perceived ownership of the project and the perceived understanding of the business requirements by the BizDevOps team. Furthermore, internal performance metrics that were encountered in the literature; such as story points/velocity, MTTR, and lead time among others are frequently in the respondents' organizations. In line with the literature, several respondents pointed towards the drawback of using velocity as a single metric for productivity, as it can be a metric that is easily manipulated and that can build rather than breakdown silo between teams.

External metrics

In contrast, external or outside-in metrics as noted by the respondents appeared to show more variability in relation to the type of industry and business model of the organization. Yet whether we are talking about a product company, IT service company or an organization with an internal IT department, several general characteristics of these metrics can be isolated. For example, user experience and external NPS are user-centered metrics that frequently appeared in both the literature and during several interviews. ROI and revenue are typical financial metrics that have also been brought up by the respondents. Moreover, usability and engagement metrics is one group of metrics that was seldomly encountered in the literature but frequently discussed during the interviews. However, this is the group that has shown the most variability due to its industry-specific nature. Despite this, the respondents have noted that usability and engagement metrics such as the number of active users and number of tickets processed are essential in providing insights into the functionality of the system and how the system is being used. This enables the teams to make fast adjustments if necessary and to benchmark, the usability towards the leading indicators set earlier on in the process. Table 5.2 on the next page outlines all metrics and their classification encountered in the qualitative study interviews:

Table 5.2
Overview of KPI metrics encountered in the interviews

Internal			
Culture & sharing		Internal capabilities	
• staff satisfaction • internal NPS	• process ownership • business understanding	• lead time of change • MTTR • rollback rate • velocity	• change fail rate • deployment freq. • error rates • focus factor
External			
User/customer metrics		Financial metrics	
• click-through rate • nr. of active users • conversion rate • retention rates • customers served • processing speed	• contact forms filled, • nr. of contracts signed • contract renewals • nr. of licenses issued • customer NPS • tickets processed	• cost-efficiency • ROI	• revenue per customer • revenue per story/feature

Before moving to the next section it is important to address the differences that were observed and identified among the participating organizations, in regard to the utilization of lagging/leading and internal/external indicators. The organizational maturity level, in this case, has shown to be an influential factor in terms of indicator usage. Organizations that were more mature in this field such as IT Monitoring Solutions and IT Stats demonstrated the capacity to combine leading/lagging and internal/external-facing indicators and channel the efforts towards hypothesis testing for correlations and value delivered. This approach was also reflected in these organizations' business value definition stage through structured value defining mechanisms that they established. On the contrary, organizations that were on a lower level of maturity such as IT Surgery Solutions and IT Fintech Platform demonstrated limited capacity in combining leading/lagging and internal/external-facing indicators. Such organizations were mostly focused on internal and lagging indicators such as internal capabilities and basic financial indicators. Consequentially, less mature organizations did not have structured value definition mechanisms but were rather exploring the best option to define and scope the value.

5.9 Continuous feedback cycles in BizDevOps

The respondents expressed a strong consensus that the establishment of continuous feedback cycles is the core of an effective, highly agile, and end-user driven BizDevOps team. Such a view on continuous feedback also aligns with the academic literature (see chapter 4). Subsequently from the interview insights, it was noted that the practices related to continuous feedback are wickedly structured. In other words, there are multiple feedback cycles in different stages of the process, on different levels, with different characteristics, and in various settings. The key is to integrate and synchronize them all together in order to ensure frequent information flow and progress updates on delivering value in line with the set business goals. IR-11 makes a distinction between qualitative feedback cycles and technical feedback cycles. Qualitative feedback cycles in this context refer to human interactions and the exchange of information during different stages of the process. Communication of defined business value, team stand-ups, and retrospectives are a few examples of qualitative feedback. On the other hand,

technical feedback cycles are related to KPI metrics and testing of software (unit, functional, integration, system, UAT). Therefore, only combined insights from qualitative and technical feedback loops would allow a BizDevOps way of working with business-IT alignment and a high level of enterprise agility, note IR-3, IR-4, and IR-11 among others.

5.10 Obstacles for BizDevOps

When asked about the biggest obstacles in front of organizations to reach BizDevOps way of working, change management, or rather the lack of it by far was the most frequent topic that was brought up. Many respondents noted that the transition to BizDevOps does not happen overnight and that careful attention needs to be given to organizational change, time and resources need to be invested in goal alignment and the promotion of agile practices. Furthermore, IR-10 notes that while BizDevOps brings innovation and novelty in making the enterprise more agile, the problem of legacy systems remains unaddressed. In other words, IR-10 stated that organizations that wish to transition to BizDevOps need to take into account whether this supports their structure and can they move away from their legacy systems. Watermelon reporting is another obstacle previously mentioned by IR-7. If there is no proper presence of continuous feedback, measuring, and alignment mechanisms, the organization runs the risk of having teams prone to deviation from the business goals and false reporting. Finally, IR-4 also made a remark on the MVP and potential dangers that Biz should be aware of. While IR-4 agrees that MVP is an effective way to move the product faster to the user to gain rapid feedback, there are also a few points for caution. For example, it is critical that the Biz does not oversell an MVP as a finished and flawless product. once the MVP gets to the user who expects a well-functioning product, an oversold MVP can fail to meet the expectations and can tarnish the organization's reputation. Moreover, it is also the Biz side to carefully lead the refinement initiative and not to approve the product too early, so that it reaches the end-user fully aligned with the initial requirements, notes IR-4.

5.11 Chapter conclusion & discussion

This chapter has summarized the insights from 13 respondents from 13 IT organizations that participated in this qualitative study. The interview transcripts were analyzed through a mix of an inductive and deductive approach, which included the use of open and axial coding to identify relevant concepts, classify them, and determine relationships between groups. The results of the analysis were summarized in a conceptual map, consisting of five core categories, two supplement categories, one outcome category, and one category representing obstacles. If the nature and patterns of the results are examined, one noticeable thing is a wide-spread consensus regarding the relevance of BizDevOps, its outcomes, the involved stakeholders, practices, processes, challenges, and problems. This at first hand may raise the questions of the qualitative study's validity, hence making it important to provide further explanation on this matter.

Several possible reasons could explain the wide-spread consensus among practitioners. First, the majority of participating organizations abide by the Scaled Agile Framework or strive to do so extensively. SAFe is a standardized and prescriptive framework with envisioned Agile processes, stakeholders, practices, and outcomes. Therefore, the implementation of SAFe as an industry-standard in participating organizations may explain their relatively shared vision of the BizDevOps approach. Another possible explanation for the consensus is the nature of the interview questions. There is a chance that the questions and the discussions have not been open enough as they might seem at first hand. Nevertheless, to further reinforce validity and minimize bias, several validation rounds of the

BizDevOps Method will be conducted where practitioners will again provide their views on the combined literature findings and qualitative study findings.

Naturally, some contrasting views have been noticed in the respondents' answers. For instance, there was a divergent opinion concerning where the role of the Business Expert should lie. Another example of a divergent opinion has been noted in the domain of performance measuring and KPI metrics, which also distinguished the way that participating organizations define their business goal and value. Several possible explanations could be drawn upon for the explanation of differing views. The most prominent explanations, in this case, lied in the different levels of maturity of the organizations, their business model, and sector of activity. Moreover, the difference in maturity, business model, and sector of the organization, for example, does not automatically mean that the insights from some organizations were more valuable than from others. On the contrary, the variation in maturity level, business model, and the sector provided enriched and multi-perspective insights. Through mature organizations, insights into best practices were gained, where through less mature organizations an equally important insight was gained into what are the main requirements and challenges on the way. Having a variety of among other product companies, IT service companies, companies with their own IT department as well as different sectors only provides a more holistic picture of industry-wide best-practices and challenges. As a result, these multi-perspective practical insights will give the BizDevOps Method more potential in regard of high-level abstraction, general applicability, and completeness.

Finally, before moving to the next chapter, it is important to shortly reflect upon the connection between the literature and practical findings, which will be synthesized in the following chapter. Based on the analysis of both chapter 4 and chapter 5 it can be stated the literature and practical insights complement and support each other. Academic and practitioner's understanding of BizDevOps relevance, outcomes, stakeholders and practices generally shows convergence in opinion. Concepts such as cross-functional teams, BizDevOps roles, continuous planning, continuous improvement, end-user centricity are just a few examples of aspects that appeared in both chapters. Furthermore, some aspects mentioned in chapter 4 were expanded and elaborated further in chapter 5. For example, practical insights expanded further on the subject-oriented approach and HCD mentioned in the literature. Practical insights provided an extended contribution to this topic through the introduction of complementary practices such as BDD, use cases, backlog management, WSJF, etc. Vice-versa, the literature insights were more thorough and holistic regarding performance measuring and validation. While chapter 5 introduced user engagement metrics as a novelty, the literature committed more efforts in providing a broader picture, thereby taking into account external/internal, leading/lagging culture, internal processes, customer and financial indicators. At last, some practices were not present or expanded in both chapters. For example, continuous BPM improvement is one concept from the literature that was not explicitly discussed in the interviews. Yet, seeing that the practitioners touched upon retrospectives, continuous learning, and improvement among other notions, an indirect relation with continuous BPM could be drawn. This concept falls under a continuous improvement mechanism, which provides grounds to take it into the synthesis of literature and practical findings which will emerge in the following chapter as the BizDevOps Method.

6. TREATMENT DESIGN

This chapter presents and explains the BizDevOps Method, along with all the steps taken to arrive at the designed artifact. In design science research, a method is an artifact type referring to a set of well-defined activities used to solve a problem and achieve a certain goal (March & Smith, 1995). A method, therefore, has a prescriptive character, meaning that it explains what to do in certain situations to arrive at a solution. Many methods are based on underlying constructs (language) and include representational guidelines (model) of the solution space. This includes procedural guidelines, for example, how to work, which steps to take, and what questions to ask (Goldkuhl et al., 1998). In the context of this research, the BizDevOps Method looks to define and describe how steps should be performed in a BizDevOps environment. It aims to prescribe rules, guidelines, and behavior patterns that can lead to a better alignment of DevOps with business goals, ultimately allowing IT organizations to maximize their agility and value delivery. But before the BizDevOps Method can be designed, it is essential to specify requirements and address any existing pre-conditions, which is what the following section does.

6.1 Requirement specification for the BizDevOps Method

6.1.1 Pre-conditions for the adoption of the BizDevOps Method

Considering that BizDevOps tries to build further on existing practices that are present in DevOps, there are several pre-conditions relevant for agile organizations to apply the BizDevOps Method. For example, if an agile IT organization intends to apply the BizDevOps Method, it is logical that the organization has a certain amount of experience and maturity with DevOps. To specify the maturity level, this thesis paper makes use of the DevOps Continuous Delivery Maturity Model (Figure 6.1), developed by Levi9 (2016).

Continuous Delivery Maturity Level	Initial	Novice	Intermediate	Advanced
Version Control	Source code	Unit tests Application configuration code	Database schema Functional tests and test data Clear branching and merging strategy Embedding traceability	Infrastructure configuration code Main branch always deployable Automated release notes
Integration	Manual or scripted builds	Scheduled automated builds (nightly) Dependency management	Continuous integration of source code and unit tests (commit based) Artifact repository	CI build cluster CI orchestrated infrastructure provisioning Continuous integration of acceptance tests
Quality	Manual testing	Test management Automated unit and integration tests	Automated functional tests Static code analysis Facilitated peer reviews	Automated acceptance tests Automated non-functional tests (performance, security) Test cluster
Deployment	Manual Manual database schema changes	Scripted Same process across different environments	Automated Deployment pipeline Gated automated promotions	Orchestrated deployments One-click deployments Continuous deployments to production
Infrastructure	Physical Manual provisioning and configuration	Virtual Highly available, redundant with fail-over	Cloud Automated provisioning Infrastructure as a Code	Self-service for experimentation Container based infrastructure Network automation
Monitoring	No monitoring	System level monitoring	Application level monitoring Log management Real-time centralized logs monitoring	Application performance monitoring Anomaly detection Dynamic scaling
Architecture	Tightly coupled	De-coupled Covered with unit tests	Service oriented Fit for purpose frameworks Testable end-to-end Enables monitoring	Component based Scalable Micro-services
Organization	Waterfall approach Silo structure	Iterative approach Direct collaboration Requirements management	Agile Lean	Multi-functional teams DevOps Technical debt addressed regularly

Figure 6.1 – DevOps Continuous Delivery Maturity Model, adapted from Levi9 (2016, p.5)

The maturity model is based on extensive research and application on major players from the e-commerce, digital marketing, and FinTech sectors to increase quality and improve time-to-market. The model contains several technical, automation, and cultural components, evaluated as initial, novice, intermediate, and advanced. For an organization to be able to implement the BizDevOps Method, it is desirable that the organization at least finds itself in the transition between the novice and the intermediary stage in the maturity model. This means that certain efforts have been made to establish CI/CD pipeline, test automation, end-to-end testing, automated infrastructure provisioning, and centralized real-time monitoring. Furthermore, the organizations have stepped away from a waterfall approach and silo structure and are applying lean and agile principles to promote cross-functional teams, iterative approach, direct collaboration with the end-user. Thus, having determined the novice-to-intermediate DevOps maturity as a pre-requisite for the application of the BizDevOps Method and having covered technical components, automation, and tooling in chapter 2 and Figure 6.1, the BizDevOps Method will not extensively cover technical aspects of DevOps related to tooling and automation. Instead, it will focus on building further by adding the Biz component to the existing DevOps chain and integrate the relevant constructs for a better business-IT alignment.

The second thing to note is that this thesis extensively uses Scrum terminology to build and explain the BizDevOps Method. Scrum is a widely used approach to structure agile development process and is commonly adopted and understood by the industry and by academia. However, while this thesis formulates the findings around Scrum, this is in no way an absolute requirement for the organizations that wish to apply the BizDevOps Method but are using Kanban, Crystal, or any other agile methodology. Scrum is rather used for convenience sake to better describe involved actors, processes, stages, and constructs.

6.1.2 General software development requirement specification

Before the BizDevOps Method can be designed, it is essential to address the requirements of the design artifact. Wieringa (2014) defines a requirement as the desired property of the artifact to achieve a particular goal. In this context, the desired goal is to enable DevOps alignment with business goals and the enhancement of enterprise agility. Forsgren (2018) for example, outlines several requirements for software product design, development, and delivery. These properties, outlined in Table 6.1 will be taken as a general requirement for the BizDevOps Method.

Table 6.1
Product design, development and delivery performance requirements

PRODUCT DESIGN & DEVELOPMENT	PRODUCT DELIVERY (BUILD, TEST, DEPLOY)
Create new products and services that solve customer problems using hypothesis-driven delivery, modern UX and design thinking.	Enable fast flow from development to production and reliable releases by standardizing work, and reducing variability and batch sizes.
Feature design and implementation may require work that has never been performed before.	Integration, test, and deployment must be performed continuously as quickly as possible.
Estimates are highly uncertain.	Cycle times should be well-known and predictable.
Outcomes are highly variable.	Outcomes should have low variability.

Note: product development, design and delivery performance requirements adapted from Forsgren (2018)

6.1.3 Requirement specification for the BizDevOps Method

Furthermore, a number of specific requirements derived from the literature and the qualitative study interviews with practitioners are taken as a foundation for designing the BizDevOps Method. To justify each requirement, this thesis provides a contribution argument as recommended by Wieringa (2014). An overview of all requirements, contribution arguments as well as from which literature sections and interview respondents they were derived are found in Table 6.2 below.

Table 6.2
Requirement specification for the BizDevOps Method

NR.	REQUIREMENT	LITERATURE	INTERVIEWS	CONTRIBUTION ARGUMENT
R1	The artifact requires a high level of abstraction and a process description that takes all relevant stakeholders into account: end-user, Biz, and IT.	Section 4.3 Section 4.8	IR-11, IR-5, IR-4, IR-10, IR-13, IR-12, IR-6, IR-3, IR-2	A high level of abstraction and process description enables a holistic mapping and understanding of the BizDevOps software development process.
R2	Biz, Dev, and Ops should be synchronized in a cross-functional team with roles and responsibilities defined.	Section 4.4 Section 4.5	IR-1, IR-3, IR-4, IR-6, IR-9, IR-10 IR-11	A synchronized BizDevOps team can achieve innovation, faster time-to-market, and stability, all while adhering to the business requirements of the end-user.
R3	The artifact should include business value definition and scoping in the process.	Section 4.5 Section 4.9	IR-2, IR-3, IR-6, IR-4, IR-13, IR-11	By defining and scoping the business value, the BizDevOps team knows the ‘why’ and has a full understanding of the business scope.
R4	BizDevOps requirement engineering should follow an approach focused on the end-user.	Section 4.5 Section 4.6 Section 4.7	IR-2, IR-3, IR-4, IR-5, IR-6, IR-11	End-user requirement engineering ensures that the BizDevOps teams gear their work towards what is being asked by the end-user.
R5	The artifact should support short and frequent feedback cycles.	Section 4.6 Section 4.7 Section 4.8	IR-3, IR-4, IR-7, IR-11, IR-13	Optimized feedback cycles enable BizDevOps teams to implement fast adjustments if necessary.
R6	Mechanisms for holistic performance tracking, measuring, and business validation should be in place.	Section 4.8 Section 4.9	IR-2, IR-3, IR-6, IR-7, IR-8, IR-13.	Such mechanisms maximize business value and enable continuous improvement.

*Note: the columns ‘literature’ and ‘interviews’ indicate the inputs used to form each requirement

R1: The artifact requires a high level of abstraction and a process description that takes all relevant stakeholders into account: end-user, Biz, and IT.

Since the BizDevOps Method focuses on the strategic alignment of business (Biz) and IT (DevOps) the prescribed set of processes require human cooperation and creativity. Rolland (1998) argues that a method with strategic implications and an emphasis on human interactions should strive to provide flexible guidance rather than strict process enforcement. Goldkuhl et al. (1998) further argue that a good feature of a method is general applicability, meaning that it is not bound to processes of a single software development company. Therefore, this BizDevOps Method places flexible guidance and general applicability central in providing a solution for a more end-user centered software development. Moreover, a high level of abstraction and complete mapping of the process would contribute to involved actors' better understanding of how the BizDevOps software development process takes place. Knowing the steps and the requirements would enable the BizDevOps team to take full ownership of the process, establish a strong systemic product thinking, thereby increasing the enterprise agility to deliver according to end-user needs.

R2: Biz, Dev, and Ops should be synchronized in an agile team with roles and responsibilities defined.

As previously noted, one of the key ideas of BizDevOps is the alignment of different stakeholders from the business and the IT domains. According to both the literature and the interview findings BizDevOps teams should be autonomous, cross-functional (T-shaped), and synchronized. The teams need to have a strongly established systemic thinking, full understanding, and ownership of the process. In this way Biz, Dev and Ops have a mutual understanding of goals and can develop an innovative and stable software product, all while adhering to the end-user requirements.

R3: The artifact should include business value definition and scoping in the process.

A key starting point to obtain a clear picture of the business requirements is to define and scope the underlying value for the business. Business value definition and scoping enable the BizDevOps team to set clear goals, know the underlying 'why' and commit an optimal amount of resources to deliver the best user experience. It is therefore imperative for the BizDevOps Method to contain business value definition and scoping aspects.

R4: BizDevOps requirement engineering should follow an approach focused on the end-user.

The fourth desirable requirement for the BizDevOps Method is to include techniques and practices that would allow the BizDevOps teams to frame the business requirements from the end-user perspective. These practices should support the Biz in grasping the end-user requirements and clearly translating the same to the DevOps team. Additionally, this requirement also extends to iterations of requirement specification as well as the management of the product backlog. Therefore, the BizDevOps Method needs to support the teams in continuous end-user requirement engineering, so that the whole process is geared towards delivering what is being asked by the end-user.

R5: The artifact should support short and frequent feedback cycles.

The presence of shorter and frequent feedback cycles in BizDevOps would make continuous learning for the BizDevOps team easier. Shorter and frequent feedback cycles enable the team to receive more valuable insights from the end-user and to make the necessary adjustments that would contribute to a better alignment to business goals. Ideally, the feedback cycles should include the following directions:

- From the end-user to Biz to DevOps and,

- From DevOps to Biz to end-user

R6: Mechanisms for holistic performance tracking, measuring, and business validation should be in place.

Performance tracking measures in place are important for two main reasons: continuous improvement and business value validation. The BizDevOps Method, therefore, has to contain a dimension that focuses on KPI metrics. According to Ravichandran et al. (2016), these metrics must be outcome/result-based, otherwise, they become vanity metrics that do not provide much insight into business performance. Furthermore, these metrics should be holistic, meaning that they should capture a variety of insights across people, process, and technology domains. Moreover, Ravichandran et al. (2016) note that the metrics should be characterized as:

- Obtainable – easy to quantify and obtain
- Reviewable – subjected and withstand business scrutiny
- Incorruptible – resistant to bias and causing internal conflicts
- Actionable – must support improved decision making

6.2 BizDevOps Method design process

6.2.1 Domain and component integration for the BizDevOps alignment

As noted multiple times throughout the paper, this research addresses the long-existing challenge of aligning business and IT domains. Just as DevOps addressed the alignment between Dev and Ops IT departments and broke down the silo between them, BizDevOps aims to do the same but between business and IT. Taking the inspiration from the Strategic Alignment Model by Venkatraman, Henderson, and Oldach (1993), Figure 6.2 represents the visualization of the desired alignment outcome. The figure specifies the link between the business and IT domains which reflects the capability to leverage business strategy, IT strategy, people, organizational and information systems infrastructure and processes to ensure internal coherence, meet requirements and delivery capabilities.

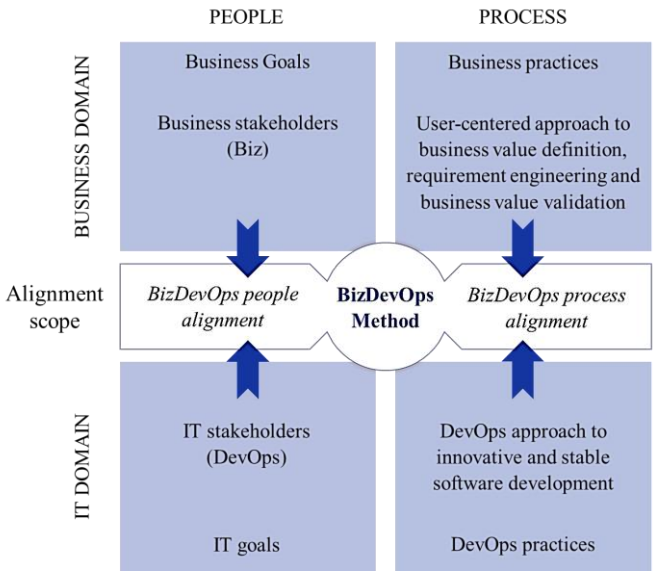

Figure 6.2 – Visualization of domain and component integration for the achievement of the desired BizDevOps alignment

The two essential alignment components that the BizDevOps aims to achieve is the alignment of people and processes. The business domain in this context refers to actors, their goals, processes, and practices that are concerned with business value definition, requirement specification, and business value validation. The business stakeholders set the business strategy and it is in their best interest to maximize value by delivering the best possible user experience. The IT domain refers to actors, their goals, processes, and practices that are concerned with the development, delivery, and operation of a software product. The IT stakeholders set the IT strategy and aim to build an innovative and stable software product efficiently. It is thus in BizDevOps where the two domains meet and where an alignment should be achieved so that DevOps is maximally synchronized with business goals.

6.2.2 Method engineering

To construct a robust artifact such as the BizDevOps Method, multiple methods, components, concepts, and practices from both business and IT domains need to be integrated and synchronized together. To unify all these elements and assemble them under one roof, the method engineering approach from Goldkuhl (1998), found in Figure 6.3 is used. Method engineering is a discipline that focuses on designing, constructing, and adapting methods, techniques, and tools for the development of information systems. Goldkuhl's method integration process was chosen for the reason that it combines different method chunks and allows the organization to ask the right set of who, what, and how questions related to the involved processes, people, and their relations.

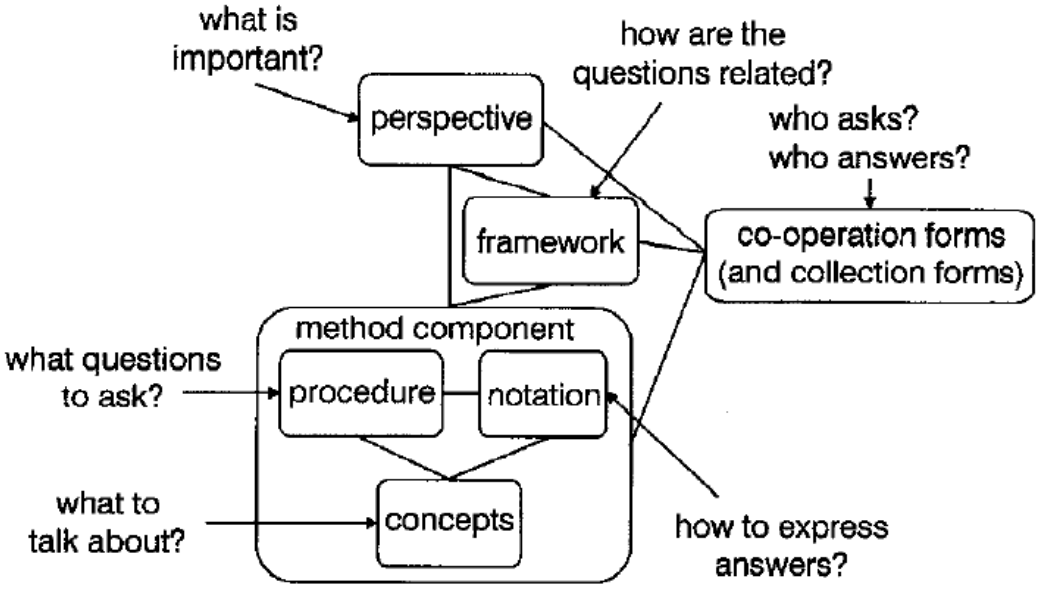

Figure 6.3 – Method engineering approach, adapted from Goldkuhl (1998, p.115)

The *perspective* in this figure represents the integration of business and IT domains (perspectives) explained in the previous section. Thus, the actors, goals, practices, and processes from both business and IT perspectives are taken into account. The *framework* box stands for the structure that is used to present (sub) stages, steps, activities, and transitions in the BizDevOps Method. The *co-operation forms* box in the figure refers to the BizDevOps team actors and definition of responsibilities, deemed necessary to function in a BizDevOps environment. Finally, the biggest box, the *method component* contains three aspects: 1) *procedure* describes the BizDevOps process flow in (sub)stages, transitions,

and concepts related to each step of the process; 2) *notation* specifies the way of expressing/annotating results in the BizDevOps Method artifact; 3) *concepts* define all necessary elements that should be integrated for the construction of the BizDevOps Method.

Moreover, Henderson-Sellers and Ralyté (2010) advocate for a situational method that can specifically address the organization's needs depending on the situation and the nature of the project. The situational method is an information system development method designed to address the context-and-project-specific situation. Henderson-Sellers and Ralyté have conducted a thorough action research study of several agile organizations and have created an agile method from method fragments. The resulting method makes a distinction between project and increment level and is depicted in Figure 6.4 where the eclipses represent high-level steps and connecting arrows the links between these steps.

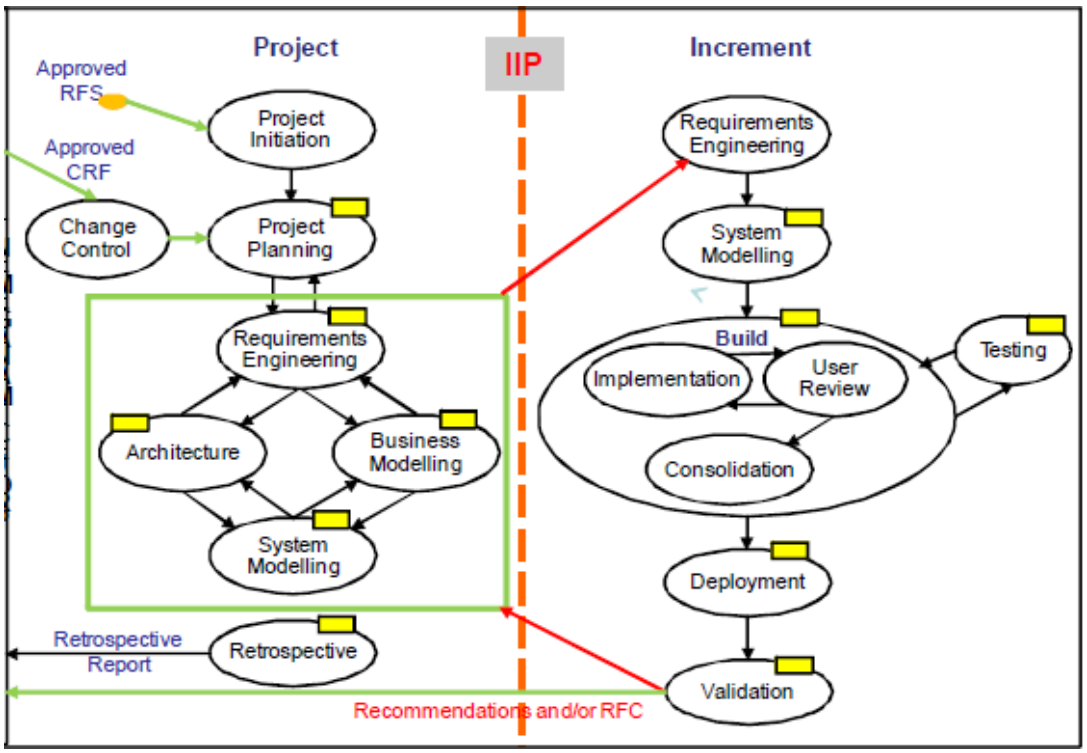

Figure 6.4 – Situational agile method, adapted from Henderson-Sellers and Ralyté (2010, p. 460)

Having addressed the requirements, pre-conditions, and method engineering practices, the following section presents the first iteration of the BizDevOps Method.

6.3 The BizDevOps Method

At last, after a thorough systematic literature study and a qualitative study, the synthesized findings on stakeholders, processes, practices, and requirements have resulted in the design of the BizDevOps Method pre-validation version, represented in Figure 6.5.

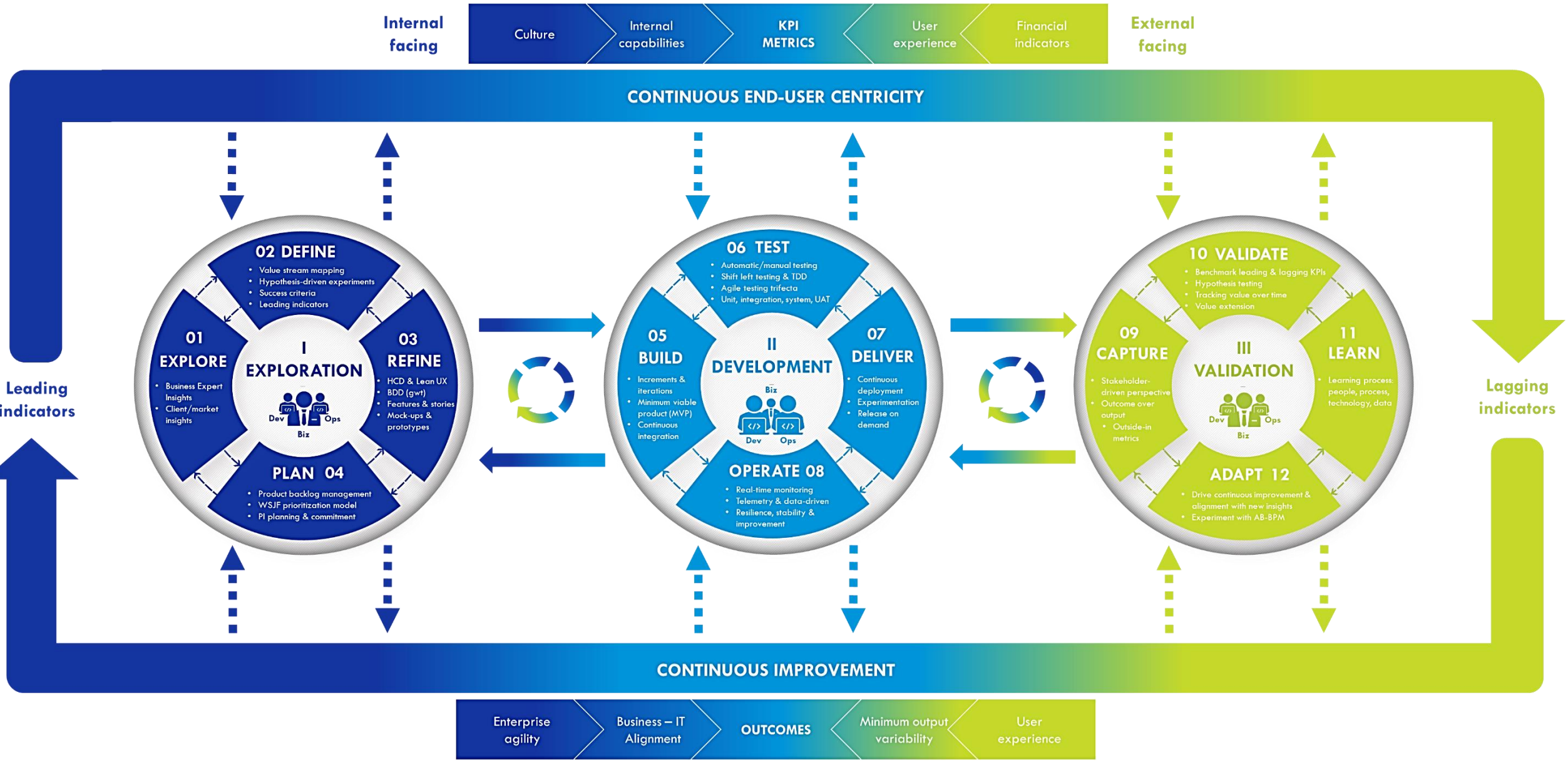

Figure 6.5 – The BizDevOps Method (pre-validation)

6.3.1 BizDevOps Method – main elements and their origin

The BizDevOps Method consists of three main, interconnected phases: 1) Exploration 2) Development and 3) Validation. The phases are connected by a set of two-way arrows indicating the exchange of information (input-output) and a cyclical arrow indicating the iterative nature of the information flow and the transition between the phases. Each phase consists of several steps that outline the essential processes and activities in that phase. The decision to structure and visualize the BizDevOps Method in these particular phases and steps stems from several sources (see Appendix D for a full overview). First, a number of literature findings have provided valuable insights into BizDevOps practices and workflow. A few visual examples that have served as inspiration include section 4.3, continuous practices (section 4.6), HCD (section 4.7), the BizDevOps process model (section 4.8), and the situational agile method (section 6.2). Second, during the conversations with practitioners from the qualitative study, several steps and phases have been derived, either directly or from context. For example, IR-11 from IT Stats specifically indicated the explore, define, refine, and plan steps in the exploration phase. IR-5 from IT Bank also directly described the workflow in BizDevOps in terms of the exploration environment, development environment, testing environment, and acceptance or production environment that corresponds to the validation phase. Finally, several frameworks such as SAFe along with whitepapers and blogs that were advised by the practitioners were also used as inspiration and reference points to visualize the BizDevOps in the current structure.

In addition, the cross-functional (T-shaped) BizDevOps team identified in literature section 4.4 and qualitative study section 5.4 is included in each phase of the process. The size of the character, corresponding to one of the roles, indicates the extent of the involvement of that role in each phase. For example, the enlarged character of the Biz role in the exploration and validation phases that Biz takes a lead role in these phases, while enlarged characters in the development phase indicate the lead role of Dev and Ops. While the artifact suggests a certain sequence of workflow, it should be noted that the current sequence is not a strict prescription, but rather a suggestion, based on the literature and qualitative study findings. The goal of the BizDevOps Method is to allow for agility in this case and enable the organizations to adjust the sequence of these steps based on the nature of their projects and processes. What is more, the steps follow the same iterative principle as the phase that they belong to, and are thus also connected in a corridor of (dashed) two-way arrows. For example, if the BizDevOps team arrived at the 'plan' step, they might wish to iterate several times and refer back to explore, define and refine steps, before they proceed to the development phase.

6.3.2 Explaining the outcomes of the BizDevOps Method

The BizDevOps Method is an approach that starts with the end in mind, represented by desirable outcomes: enterprise agility, business-IT alignment, minimum product variability, and user experience. In other words, the artifact aims to provide guidelines for organizations to enhance their ability to adjust to changing business needs, align their processes with business requirements, and gear development and delivery towards maximizing user experience. These desirable outcomes were derived from the literature findings in chapter 4 (sections 4.3, 4.6, and 4.7) and interview findings that were visualized in section 5.2. To enable the achievement of such outcomes, the BizDevOps Method puts a central emphasis on iterative thinking, information flow, and the presence of continuous feedback cycles, alignment loops, measuring, and end-user centrism. As a result, these aspects serve as a continuous improvement mechanism that enables organizations to iterate, evolve, and maximize end-user value. The above-mentioned elements are visualized by different types of multi-directional and multi-level arrows, which serve to indicate a feedback point, information exchange, or iteration. These enabler

insights were identified both by the literature sections 4.6 – 4.9 and the interviewed practitioners (see the conceptual map in section 5.2).

Now that the main elements and outcomes of the BizDevOps Method have been defined and explained, it is time to dive deeper into the artifact and explain each component in detail.

6.4 Phase I - Exploration

The purpose of the exploration phase (Figure 6.6) is to identify market needs, turn them into innovative ideas, define high-level business goals, specify requirements, translate these into product design and plan the development. Therefore, it is a pre-development phase that aims to align business goals with development and gear the whole process towards the defined business goals and the end-user. The exploration phase is an essential first alignment point between Biz and DevOps because it enables organizations to start with the end in mind, to ask the right set of questions, and to set proper requirements based on what the business wants rather than technical capabilities. Through a proper value and requirement definition, the exploration phase aims to minimize the chance that the BizDevOps team experiences rollbacks and other surprises later on in the process that could jeopardize the ability to deliver maximal value to the end-user.

Figure 6.6 – Exploration phase of the BizDevOps Method

It should be noted that the amount of effort and resources that an organization should invest in the exploration phase, largely depends on whether the software development concerns a new or an existing product. In the case of the former, the organizations can expect to spend more effort in the exploration phase, as many factors are still unknown and higher efforts are needed to define problems, goals, ideas, requirements and come up with a viable solution. This also means higher involvement of the BizDevOps

team, extensive communication with the customer, idea conceptualization, and feedback and sessions. In the case of the latter, where the organization already has a working software, the level of uncertainty is significantly lower and thus the exploration efforts are expected to be lower. A number of requirements have already been defined and translated into working products. The emphasis, in this case, lies more on the improvement and iterations of the existing product, adding upgrades, extensions, and maintain stability in the process. However, maturity does not mean the end. It is rather a process of continuous improvement, iterations, and extended value creation for the end-user.

6.4.1 Explore

The first phase of the process is initiated by the exploration of the client's problem, needs, and market insights to identify new opportunities, innovate, and create value. In this step, the software development organization gathers the information from multiple sources to gain an understanding of the situation. In this case, the Business Expert serves as a valuable source of input. As previously noted from the literature and the practical findings, the Business Expert has a varying role, depending on the organization's business model and sector. This means that several stakeholders such as the client, sales engineer, the key-user, business analyst, and success managers provide valuable input on the latest market trends, business problems, and needs.

6.4.2 Define

Once the organization has gained a good understanding of the underlying problem and market needs, the next step is to define high-level business goals. Usually, high-level business goals are defined on the level of senior management and it is the task of the Product Manager together with the assistance of the Product Owner and Business Analyst to set high-level objectives, define the vision and decide on what is a viable product to build. The value definition step must take the whole value stream into account, including people, process, technology, and data (IR-2). The process of value definition is a hypothesis-driven and experimental approach. Namely, a hypothesis serves to define success/failure criteria of a business outcome that can later on in the process be tested and validated with brainstorming and end-user feedback. Therefore, a hypothesis serves to evaluate whether the end product in use is delivering the value promised to the end-user. As previously noted by IR-3:

> *"You need to understand how does success but also failure look like. If you know what the business goal is, you can translate it into something that you can measure and what you can measure, you can evaluate and adjust based on the feedback that was received."*

The value definition process also includes the definition of leading indicators, which play an important role as they provide inputs into what actions are necessary to achieve certain goals. Moreover, leading indicators serve as a benchmark to validate later on in the process whether the end product is meeting the defined business objectives. From the literature and practical findings, business impact mapping and product roadmaps have been derived as prominent tools that are used to scope the business value, define leading indicators and communicate solution deliverables to all involved stakeholders.

6.4.3 Refine

Following the principles of Human-Centered Design (HCD) and Lean UX the Product Owner, usually supported by the Business Analyst, Business Expert, and lead Dev works on breaking down high-level business goals and translating them into features, stories, and use cases. HCD and Lean UX principles,

enable the BizDevOps team to iteratively frame the problem, specify requirements, come up with a mock-up or a solution prototype, evaluate it with the client, and receive rapid feedback. A prominent technique and best-practices for the requirement engineering process, derived from the practical findings is Behavior-Driven Development (BDD). It is a scenario-based approach that entails the description and documentation of possible scenarios of user's behavior while using the software product. Once the behavior and outcomes are described, acceptance criteria on the feature level can be set, for example using the given-when-then framework (SAFe, 2020). An example of a GWT scenario could be:

- *Given* that we build a new payment function
- *When* the user makes a transaction of €100 to receiver A
- *Then* the system should process the transaction and deliver €100 to receiver A

6.4.4 Plan

Finally, in the planning step the BizDevOps team estimates and plans the product increment that is to be built in the following period. A general continuous planning practice found in many organizations is the PI planning event, which serves to bring all teams and stakeholders involved in the development of a specific product. PI events serve as an essential alignment loop to create a unified vision of the product features, planning, team dependencies, and other necessary components to end-user value. To do so, the product management and the Product Owner must define and prioritize the backlog in coordination with the rest of the team. The product backlog management is a continuous and incremental process with constant updates and it exists on several levels of granularity; from high-level solutions to features that address business needs and team-level user stories on which several BizDevOps teams work to deliver the features. In the setting of the team backlog, the Product Owner is responsible to define and prioritize what gets built in the upcoming product increments. Weighted Shortest Job First (WSJF) is a notable model for backlog prioritization that helps to ensure the synchronization and alignment of features/stories with the defined business goals and end-user requirements. WSJF model enables product teams to effectively decide what gets built first by looking at the business value for the user, time criticality, risk, opportunity, and estimated duration (size) of the project. Consequentially, the BizDevOps teams should focus on the jobs that deliver the highest value in the shortest amount of time. At last, once the prioritization of the team backlog is determined by the Product Owner, the BizDevOps decides which stories they will commit to building in the upcoming product increment.

To sum it up, if the BizDevOps teams feel comfortable with the explored ideas, defined goals, specified requirements, client evaluation, and feedback from the prototype, the transition to the next phase of the process can take place.

6.5 Phase II - Development

Having explored, defined, and aligned goals and value requirements in the exploration phase, the BizDevOps team now has a solid foundation and input on functional and non-functional requirements to start developing the software product. Contrary to the exploration phase where the Biz roles, the product management, Product Owner, BA, and Business Expert had the leading role, in the development phase, DevOps takes the lead in developing and delivering the software product. Business functions in this phase support the process, thereby serving as a communication line with the end-user, providing additional input on the changing business needs, managing and aligning the involved stakeholders and processes. Scrum Master/Agile Coach is a good example of a supportive role who promotes

collaboration within the team, helps the teams in coordination, eliminate bottlenecks, rule enforcement, and promotion of agile practices. As previously noted, this thesis paper mainly focuses on providing new insights related to the Biz extension of the DevOps chain rather than technical insights related to automation and tooling in the DevOps domain. Nevertheless, this thesis paper takes the technical insights into the discussion for the development phase from the perspective that looks at how to bridge business and IT. Figure 6.7 provides the visualization of the development phase in the BizDevOps Method.

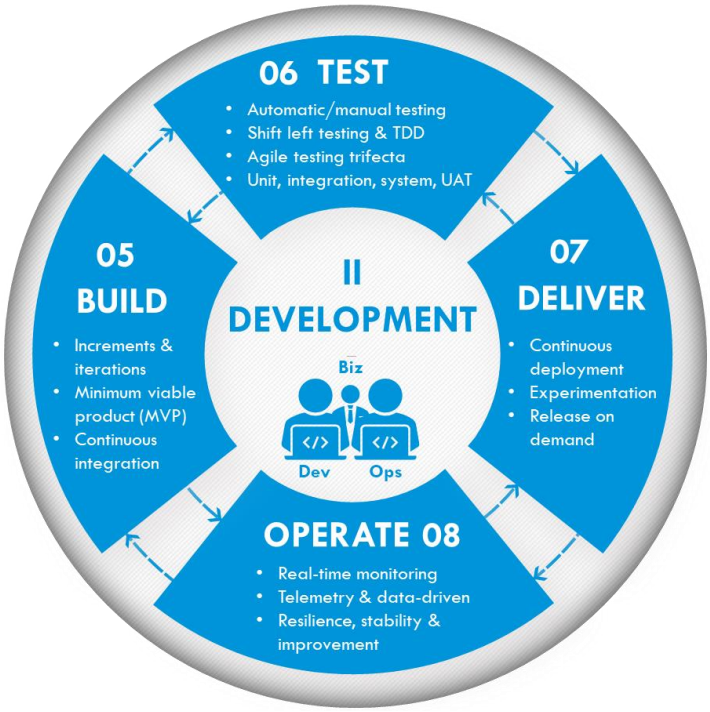

Figure 6.7 – Development phase of the BizDevOps Method

6.5.1 Build

In the building step, the BizDevOps team employs continuous integration (CI) practices to develop, test, (automatically) integrate code and validate the product increment which would enable safe and satisfying deployment into production. Since in this process the BizDevOps team takes an incremental approach, it means that several increments and iterations will take place, depending on the complexity of the end-user needs. During the product increment rounds, the BizDevOps team is expected to have multiple meeting sessions, daily stand-ups, sprint reviews, and sprint retrospectives where they can reflect on the process so far and make adjustments where needed. Usually, in the first round of the product increment, the team works on turning the prototype into a working version of the software, also known as the minimum viable product (MVP). The goal of an MVP is to develop a version with minimal working requirements as fast as possible in order to obtain maximum input from the end-user and work further on improving the software.

6.5.2 Test

Once the codes have been integrated into a build using continuous deployment, the software is moved into the testing environment where depending on the maturity of an organization a set of automatic and manual tests takes place. The software goes through a set of rigorous tests such as unit testing,

integration testing, system testing, and later on user acceptance testing (UAT). UAT testing is usually done by the key user and the client (not necessarily the end-user) in a pre-production environment An important phenomenon in testing is the concept of Shift-left testing, which emphasizes testing early on in the process and test automation, test data management, and test constrains removal (Agile testing trifecta) to resolve defects early and accelerate time-to-market. A fundamental enabler of the shift-left principles is test-driven development (TDD) which follows the philosophy that recommends developers to write and execute automated test cases to verify the piece of code or system component that is yet to be implemented. Along with the previously mentioned BDD, TDD is an effective way to embed quality into development through the test-first principle (SAFe, 2020).

6.5.3 Deliver

Once the testing takes place, the team is ready to bring the validated software features into a staging environment, preferably with continuous deployment and continuous delivery (CD), where the product increment can automatically be released into production on demand. In this case, one should make a distinction between deployment and release in the sense that the former refers to software features being brought into a staging environment but not necessarily available to the end-user, which is the case in the latter. A staging environment is the final point of testing and approval before release. Eventually, when the new product feature is released and introduced to the client, the CD principle promotes experimentation with the targeted group of users in production to quickly validate the new features, compare the outcome with expectations and make informed improvements. Some prominent techniques used in this situation are the previously covered A/B testing and hypothesis-driven development. Additionally, the BizDevOps team can resort to canary releases where the change is gradually introduced and tested to reduce the risk of bugs. Blue-green deployment is another method of testing out the software that allows the team to deploy changes into two production environments, one that is live and one that is idle (Posta, 2015). In this way, the team can use the idle environment; for example, blue for staging and testing while the green environment is live. Once the new changes are ready for release, the team can switch the router and make the blue environment live while green becomes an idle environment where the repeated process of staging and testing of new changes can take place. Blue-Green deployment enables rapid rollbacks by switching back and forth between the environments (Posta, 2015).

6.5.4 Operate

As the software product finds itself in the production environment and is in use by the end-user, it is up to the Ops to establish the mechanisms for centralized and real-time monitoring of the application in use. Monitoring of the application in this sense serves as a technical feedback loop where telemetry data (events, logs, metrics) is gathered with monitoring servers. Telemetry simply refers to the process of recording the behavior of the system and is an important aspect for maintaining system stability and locating space for improvement. Telemetry data is used to gather the data referring to the application health, usage, business performance, environment, and deployment pipeline among others (Solajic & Petrovic, 2019). BizDevOps teams should see the use of telemetry as a disciplined data-driven (factual) approach to detect problems and bottlenecks in the applications that would allow the team to adjust or even pivot towards a different solution. It is important to note that the BizDevOps team should narrow down their focus on solving the biggest problems; the so-called ‘million-dollar problems’, not the ‘five-dollar problems’. As noted by one of the interview respondents:

"I would say that your first feedback is not personal feedback, but is in metrics. If you have good metrics, you can see how your system is used, and you can put good monitoring on it. If you always have 1000 views per day on a certain page or at the end of a certain funnel, and all of a sudden, you have 100 or zero, you want to get notification from that, you want the big red alarm to go off because something is probably wrong. So it's up to you to build in, those checks and balances." (IR-4)

The bottom line of the operation step is that the Ops should gear the software operation by the motive to maximize the value, quality, and help the end-user to achieve desired goals. How should the BizDevOps team confirm whether their efforts to deliver value have been successful, is something that the following phase of the BizDevOps Method examines.

6.6 Phase III - Validation

The final phase presented in Figure 6.8 essentially provides the means to evaluate whether the delivered product is meeting the end-user needs. In the validation phase, it is again the Biz that takes the leading role in examining the end-to-end performance and assessing whether the end-user received the promised value. The main purpose of the validation phase is to test and validate previously stated hypotheses by benchmarking the defined business goals and metrics (exploration phase) with delivery outcomes and results. This process enables the BizDevOps team to reflect on the whole software delivery process, to define the outcome optimum and close the case, or to learn what refinement steps are needed to reach the state of optimum. The validation phase is thus a way for the BizDevOps team to confirm, adapt, learn, and improve.

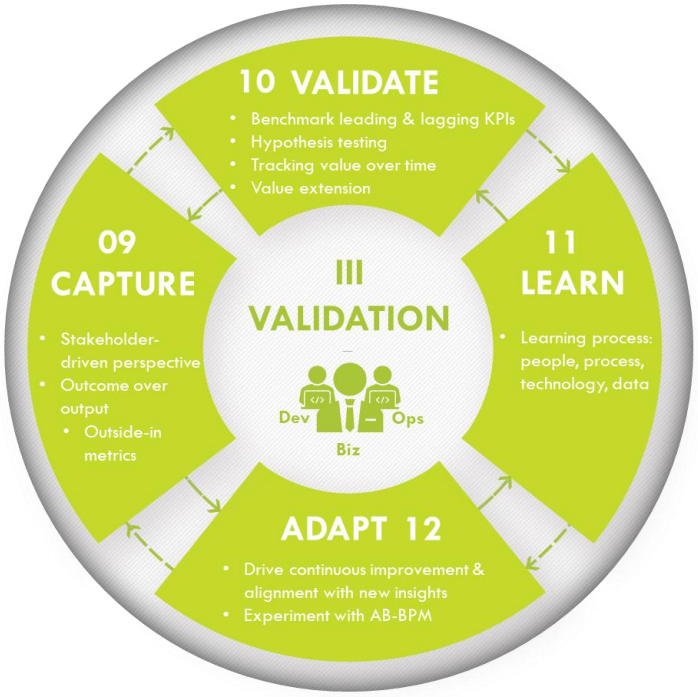

Figure 6.8 – Validation phase of the BizDevOps Method

6.6.1 Capture

After collecting data in the telemetry process, the first step in the validation process starts with isolating and deriving the indicators that capture the business value delivered. There are several important principles concerning the type of indicators that this paper has identified from the literature and practical study. The BizDevOps teams must take a stakeholder-driven and value chain perspective in assessing performance. Both the academic literature and the practitioners have shown a strong consensus in advocating the focus on outcomes (results) over output. Value-focus in this situation is paramount for a BizDevOps team. Measuring business impact with a feature as output is simply difficult to measure and it is not certain that that feature specifically increases for instance revenue. It is therefore important for the teams to classify and prioritize KPI metrics according to primary performance objectives, secondary performance objectives, and performance enablers. User experience-related metrics and financial metrics are commonly recommended metrics to be categorized according to primary performance objectives as they are seen as outside-in measures of ultimate value delivered, external impact, and outcomes. Furthermore, it is recommended to categorize metrics capturing internal capabilities (efficiency, quality) according to secondary objectives as they are a reflection of the internal process in respect to delivery and thus improvement leverage. Finally, culture metrics should be taken as the performance enablers that allow for innovation and measurement capabilities for performing.

6.6.2 Validate

In the validation step, the BizDevOps team examines how the targeted KPI measures (leading indicators) defined in the exploration phase are holding after the software has been used in production by the end-user. At this point, it is recommended that the BizDevOps team benchmarks the forward-looking (desirable) leading indicators and lagging indicators of current performance. How these indicators are specified on the unit level is something that has proven to be dependent on the organizational operating sector, business model, and maturity. Therefore, metrics are expected to vary across software product companies, IT service companies, and government for example. Nevertheless, the general pattern of practices recommended in the previous section did prove to be generally applicable. As such Figure 6.9 on the next page represents a visualization of BizDevOps performance measuring. The figure also outlines several most commonly encountered unit-level metrics per category, encountered in the literature and the qualitative study.

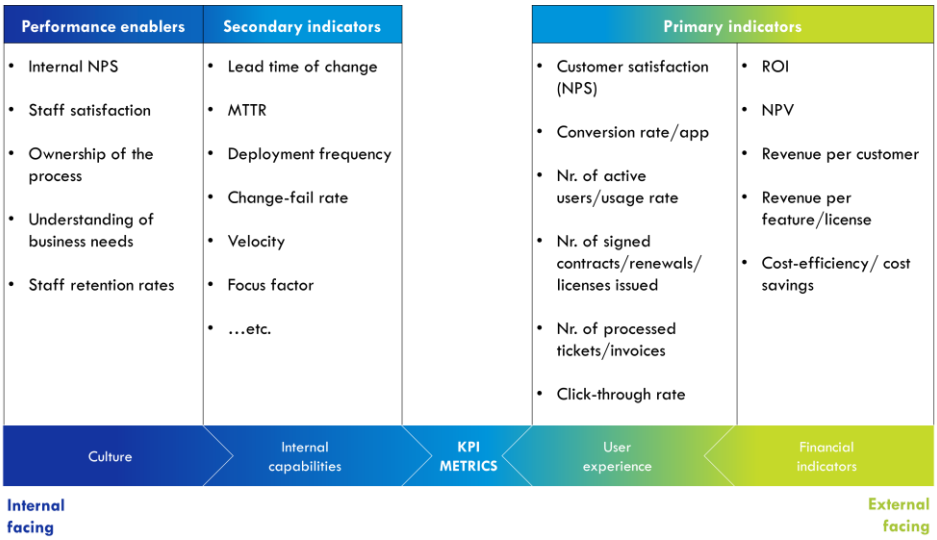

Figure 6.9 – KPI metrics in BizDevOps Method

It is furthermore suggested that the BizDevOps team tests the hypothesis, stated in the exploration phase to validate whether the outcome is satisfying the defined success criteria. In other words, has the software product been delivered in an optimum state? Moreover, testing whether or not a satisfying level of value has been delivered is a process that might require a certain period of time. For example, it might take a few weeks for the software to scale and show certain outcomes. In any case, monitoring developments and validating delivered value is something that is recommended to be done over time. Business Experts and Customer Success Managers could in this instance serve as an extension for further value maximization.

6.6.3 Learn & Adapt

After validating business value delivered to the end-user, the BizDevOps Method advocates continuous improvement through the application of lessons learned and adaptation across the value stream and on different levels of the organization. Having applied the iterative approach with rounds of product increments, intensive testing, and validation, the organization implementing BizDevOps gains insights across the exploration, development, and validation phases of the software delivery process. New knowledge on best practices in terms of team coordination, delivery process, technical capabilities, end-user engagement with the software, and new market trends can serve as a part of a learning process to drive a continuous improvement loop and tighter business-IT alignment. In order to test whether the made adjustments are bearing results, the organizations could experiment using the principle of AB-BPM mentioned by Satyal et al. (2019) in the literature. AB-BPM is similar to A/B testing in the sense that it tests two or more different versions on the process level. This for example implies that the organization could generate two or more versions of adapted software delivery processes, assign them to different BizDevOps teams and run them parallelly. The outcome of such an approach should lead to an ultimate convergence towards the best performing version of the process. AB-BPM is thus one of the possible ways to drive and enhance continuous learning and continuous improvement.

6.7 Chapter conclusion

To sum it up, this chapter presented and explained the BizDevOps Method, along with all the steps taken to arrive at the designed artifact. It has been found that for organizations to adopt BizDevOps a certain level of DevOps maturity is required (predominately intermediate or higher). Several requirements, among which high-level of abstraction and general applicability have been defined as integral to the BizDevOps Method in section 6.1. In addition, section 6.2 indicated that in order to construct a robust artifact such as the BizDevOps Method, multiple components, stakeholders, concepts, and practices from both business and IT domains need to be integrated and synchronized together into a context-and-project-specific situational method. The remaining sections in this chapter were reserved for the visualization and the explanation of the 3-phased BizDevOps Method. Iterative thinking, information flow, the presence of continuous feedback cycles, alignment loops, performance measuring, continuous improvement, and end-user centrism among others, have been identified as critical components of the artifact and the enablers for the achievement of desirable outcomes. These outcomes include enterprise agility, business-IT alignment, minimum product variability, and user experience. At last, now that the BizDevOps Method has been designed and explained, what remains is the validation of the artifact which is what the following chapter will cover.

7. TREATMENT VALIDATION

This chapter presents the approach and results of the validation rounds regarding the designed artifact, the BizDevOps Method.

7.1 Validation approach

The BizDevOps Method went through two formative evaluation rounds where a group of experts evaluated the artifact on a number of defined criteria. As noted in chapter 3, a formative focus group was utilized as a technique to discuss and collect expert opinions to make improvements to the artifact and uncover possible limitations. Tremblay et al. (2010) provide several arguments for the appropriateness of a focus group as an evaluation technique for DSR:

- focus group allows direct interaction with the participants,
- provides flexibility in dealing with various design ideas and
- offers a way to generate new ideas from participants' comments

The two formative evaluation rounds were conducted in cooperation with two organizations. Due to the current circumstances surrounding the pandemic, both sessions were held using an online panel presentation and discussion. The first validation round was conducted with the consultancy and technology company, Quint, where the BizDevOps Method was presented to five medium-senior level IT consultants, specializing in digital strategy, agile transformation, and business-IT alignment among others. In the second evaluation round, the BizDevOps Method was presented to the representatives of the software engineering company, Levi9 and several interview respondents from the qualitative study. The total number of people that attended the second round was eight, with diverse Biz and IT roles as well as a varying level of experience in their respective fields.

Regarding the organizing aspects of the focus group sessions, all participants have received informative materials before the sessions, including the validation protocol. Each participant has beforehand received the description and visualization of the designed artifact, the BizDevOps Method. Additionally, presentation slides, agenda, and discussion points have also been provided, so that the participants could prepare. During the session, a presentation was held to introduce and elaborate on the BizDevOps Method. Afterward, the session moved to a critical discussion segment where the participants were able to provide their feedback based on a set of open-end questions. Both, the sessions with Quint and Levi9 lasted approximately one hour each.

After the two validation rounds with Quint and Levi9, the feedback from the respondents was used to make incremental improvements to the artifact. Finally, as a part of the summative evaluation, all respondents received the modified version of the BizDevOps Method and were asked to fill in a short survey. This can be regarded as the third (summative) evaluation round and it consisted of five closed questions on a scale from 1 to 5, based on the pre-defined evaluation criteria and one open question for additional comments and suggestions.

7.2 Evaluation criteria

To validate the BizDevOps Method consistently, a set of evaluation criteria was selected based on the work of Alturki et al. (2013) and Prat et al. (2014). Both authors looked into the most appropriate evaluation criteria for DSR and have identified three common criteria: 1) understandability, 2)

completeness and accuracy, 3) usefulness and efficacy. On top of these three criteria, Prat et al. (2014) identified organizational fit as the fourth common criteria. Prat et al. (2014) furthermore, extensively examined 26 research papers and derived a hierarchy of evaluation dimensions, criteria and sub-criteria for IS artifact evaluation. Out of all identified criteria, only six have been utilized more than twice, as can be seen in the figure below.

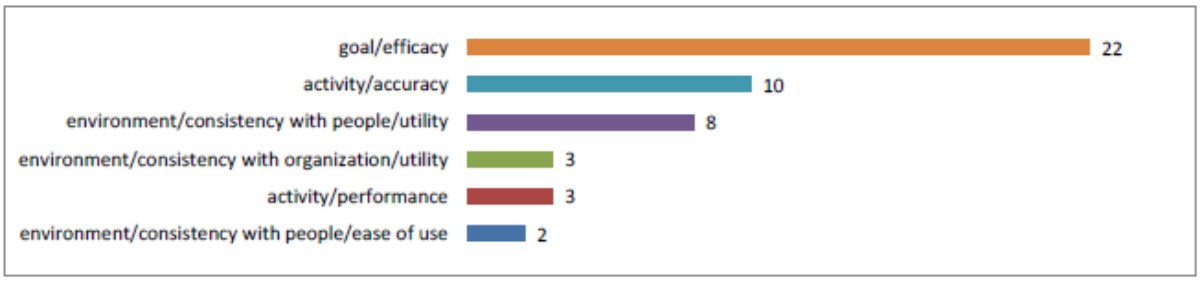

Figure 7.1 – DSR evaluation criteria frequency of use in twenty-six papers identified by Prat et al. (2014)

The usefulness of the artifact or goal/efficacy as classified in the figure is by far the most frequently used criteria and is therefore selected for the evaluation of the BizDevOps Method. Moreover, from the six criteria in the figure, three of them with the environment dimension can be grouped and classified as the organizational fit, while the two criteria with the activity dimensions can be grouped as completeness and accuracy. Finally, while understandability is not included in this figure, Prat et al. (2014) and Alturki et al. (2013) recognized its descriptive and explanatory relevancy, which is also why this paper includes it into the evaluation of the BizDevOps Method. The description of the selected criteria for the validation of the BizDevOps Method is given below:

- **Understandability:** the first criterion assesses the clarity of the BizDevOps Method and aims to uncover if the given names, concepts, visuals are straightforward and intuitive.
- **Completeness and accuracy:** this criterion is used to assess whether the BizDevOps Method accurately covers all the necessary steps and is not missing any components. It therefore, aims to obtain the evaluator's opinion of the correctness and completeness of the BizDevOps Method.
- **Usefulness and efficacy:** the third criterion seeks to examine to what extent does the BizDevOps Method address and solve the intended problem; the gap between business and IT. Therefore, usefulness and efficacy look at whether the artifact adds value to the organization.
- **Organizational fit:** since the artifact is aimed at having a high level of general applicability, this criterion seeks to examine the applicability of the BizDevOps Method in the organizations of the participants. An example of aspects that this criterion takes into account is the level of organizational maturity and fit with the people.

Table 7.1 on the next page summarizes the approach to BizDevOps Method validation, including the validation strategy, validation technique, formative-summative distinction, participants, and the utilized valuation criteria.

Table 7.1
Summary of the validation approach

COMPONENTS	DESCRIPTION
Validation strategy	Human Risk & Efficiency (Venable et al., 2016)
Validation technique	Expert opinion (Wieringa, 2014)
Formative evaluation	Two validation rounds in the form of exploratory focus groups – presentation of the artifact and open discussion for improvement.
Summative evaluation	Quantitative survey with 1-5 scale questions based on pre-defined evaluation criteria to rate the final artifact.
Participants (13)	Quint consultants (5), Levi9 Technology Services (4), interview respondents (4)
Evaluation criteria	Understandability, Completeness & Accuracy, Usefulness, and Organizational fit

7.3 Formative evaluation results

7.3.1 Understandability

The experts generally agreed that the BizDevOps Method was comprehensive and easy to understand. The flow of the artifact and its components were logical and intuitive for the respondents as well. Yet, at the same time, the BizDevOps Method was experienced as a rather complex artifact which leaves several implications in terms of presenting the artifact to organizations. For example, the experts suggested accommodating the presentation of the artifact and the level of detail, depending on the knowledge and experience of the audience. Likewise, it was noted that the BizDevOps Method as-is, has enough flexibility to accommodate to knowledge and experience levels. In addition, the experts indicated that the organizational problems usually are located at specific steps and phases that the BizDevOps Method outlines, meaning that the presentation of the entire artifact would sometimes not be needed. It was therefore suggested to consider highlighting certain parts of the artifact depending on the organizational problem that is being addressed. Thus, the suggested points of improvement in dealing with the complexity of the BizDevOps Method is a matter of approach to presenting the artifact to different audience and organizations. Therefore, no specific changes to the artifact have been identified as necessary based on this criterion.

7.3.2 Completeness and accuracy

The experts considered the BizDevOps Method as partially complete. The participants indicated that the artifact is relatively holistic and that it mostly covers the steps that are seen in practice. Yet, several points for improvement have been suggested.

The first major point for improvement was to indicate more thoroughly what is the role of Biz in the development phase and even to change the name of the development phase into the BizDevOps phase that is the core. According to the participants, the role of Biz in this phase is to direct the team in focusing on the highest value requirements. The Biz, for example, participates in UAT testing, product demos, and delivery. For example, the Business Analyst is the one that knows the functional requirements and how the system should work, which allows the Biz to validate the value delivered. The Biz is also responsible for supporting training efforts, answering the user's questions, collecting

additional requirements, and communicating those back to the BizDevOps team. Finally, the Biz sometimes participates in the operations part by helping in the assessment of the severity of defects, finding the solution, and mitigating risk.

Another improvement point was that the BizDevOps Method was to take the implementation into account. While it was noted in the previous chapter that validation is a process that takes time, thereby indirectly referring to the factor of implementation of the software, the experts noted that implementation is an important aspect in this case and should be taken into the consideration. Nevertheless, taking the scope of this research into account, it was suggested that it would be good to at least visually include implementation in the validation phase. Also, it was suggested to find a way to visualize the end-user in the artifact, since the user is the central focus of the BizDevOps Method.

7.3.3 Usefulness and efficacy

The BizDevOps Method was evaluated as effective in helping organizations to enhance enterprise agility, business-IT alignment, and establishing end-user centricity. The respondents express the artifact's usefulness in helping organization visualize how a holistic transition towards BizDevOps should look like. With a good overview of steps, practices, and stakeholders, the BizDevOps Method is deemed as a useful tool to set the stage and guide the BizDevOps transition, communicate points for improvements and goals as well as to keep the focus on them. Quint consultants for example noted that the classification (leading/lagging and internal/external) and prioritization (primary, secondary, and enablers) of KPI metrics, as well as the identification of the frequently used metrics, offers a lot of value to organizations in setting and keeping the course in the BizDevOps transition. The experts furthermore suggested that it would be helpful to identify what are the topmost occurring problems across organizations and to classify, prioritize, and attach a set of KPIs around targeted problems.

Another point that was made is that while the BizDevOps Method focuses on describing and prescribing the end-user centric approach that includes Biz, Dev, and Ops working in the same team, little attention has been given to the question of how to structure the teams and the organization. In other words, the respondents indicated the need for BizDevOps Method to investigate structural implications and to examine the artifact also on a strategic and tactical level. Finally, several respondents acknowledged the artifact for stressing an iterative approach and frequent feedback cycles. However, they indicated that in practice, it is important to strike a balance in the intensity of iterations and feedback cycles, thereby insisting that changes should be avoided during the sprints, as it allows Dev to focus on finishing the increment.

7.3.4 Organizational fit

The experts regarded the BizDevOps Method as generally applicable but have also raised several comments regarding the fit with the organization. The participants agree with the notion of organizational maturity that was previously mentioned in section 6.1. According to the experts, the artifact is likely to help organizations in making progress towards BizDevOps. Nevertheless, they deemed the artifact challenging to implement in its entirety, as only the most mature organizations possess the needed resources and agility to completely implement the BizDevOps Method. What is more, the experts experienced the BizDevOps Method is an artifact that would work well for software product companies (commercial use) while for companies that are building a system for an internal department such as logistics, some parts of the artifact are less applicable or less relevant. In this way, the experts indicated that while the artifact seems to be generally applicable, its relevancy and fit will

vary depending on the business model of an organization and the purpose (commercial or internal use) for which the software is being built.

7.4 Artifact modification after the formative evaluation rounds

The findings from the validation rounds indicate that several changes to the BizDevOps Method are needed. Therefore, the evolution of the artifact contained the following changes:

- **Changing the actors:** as proposed small-scale and cross-functional BizDevOps team were emphasized more and several roles that are typically a part of the BizDevOps team have been added. These roles have previously been encountered in the literature, qualitative studies and were mentioned during the validation rounds. Additionally, the end-user was visualized in the main alignment loop as suggested by Quint consultants, as the end-user is the central focus of BizDevOps.
- **Renaming the development phase and emphasizing the Biz role:** according to the participants it was suggested to change the development phase into BizDevOps in order to emphasize the core, which is the alignment of Biz and DevOps. Moreover, the role of the Biz in test, support, and updating has been added in this phase. Yet, while further visualization of the Biz role in this phase was not feasible, the elaborated explanation of the responsibilities was provided and incorporated in section 7.3.2.
- **Emphasizing implementation in the validation phase:** tracking value over time was replaced by implementation in order to more directly stress the influence of the implementation process on the time that it takes to validate the business value delivered. How this implementation process should look like is not within the boundaries of this research.

Other mentioned points for improvement such as the problem-targeted KPI metrics and the structural implications of the BizDevOps Method are not within the scope of this research, and could thus not be incorporated into the BizDevOps Method evolution. This thesis paper strived to develop a method that has a high level of abstraction and one which equips organizations with enough flexibility to tailor the artifact to their specific internal processes and people. The mentioned points of improvement do however provide valuable insights on the limitations of this research and set the ground for further research, something that is discussed in the following chapter 8.

The final version of the BizDevOps Method after the formative evaluation rounds can be found in Figure 7.2 on the next page.

7.5 Summative evaluation results

At last, the new version of the artifact has been sent to the experts for the third and final (summative) evaluation round. As noted by Venable et al. (2016) summative evaluation is used to measure the results of completed development and to judge how well the outcome matches the expectations, which in this case concerns the final version of the BizDevOps Method. The authors further argue that summative evaluation focuses on a rigorous evaluation of the artifact's effectiveness. In other words, summative evaluation in the context of this research is used to examine to what extent will the utility/benefit of the artifact continue to accrue even when the BizDevOps Method is placed in operation in the real organizational context and over the long run, regardless of the human adoption challenges.

BIZDEVOPS METHOD

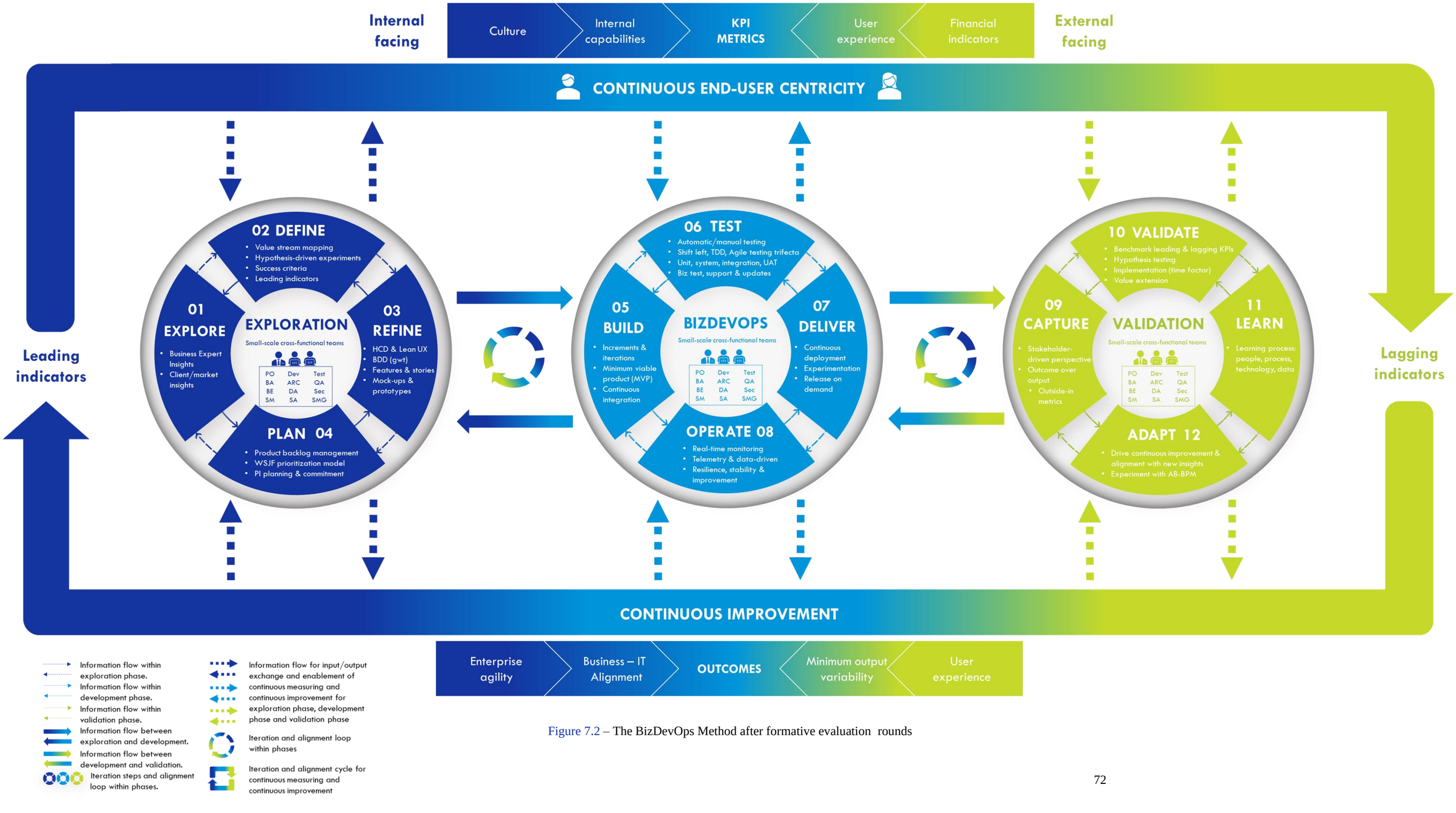

Figure 7.2 – The BizDevOps Method after formative evaluation rounds

The experts have received a small evaluation form containing five scale questions (1-5) and one open question. Four out of five scale questions have been based on the previously stated evaluation criteria while the fifth scale question and open question allowed the experts to indicate their overall impression of the artifact and additional comments. The evaluation form has been filled by 11 experts and the average result for each question is found in the figure below:

Figure 7.3 – Summative evaluation round results

The experts on average evaluated the BizDevOps Method on a higher-end, as having the solid potential to provide sufficient utility/benefit to their and other organizations if applied in the organizational context. Their overall impression of the artifact was scored 4.2 out of 5. The majority of comments referring to improvement and limitations of the artifact overlapped with the improvement points mentioned during the previous two evaluation rounds. Structural implications for enterprise governance and difficulty in a holistic implementation were some of the repeated comments.

7.6 Chapter conclusion

To summarize, this chapter presented the validation approach and the findings of the validation rounds that were used to gain insights on possible improvements and limitations of the BizDevOps Method. Two focus group sessions with 13 experts have been conducted as a part of the formative evaluation to discover improvement points and possible limitations of the BizDevOps Method. The artifact has been evaluated on understandability, completeness and accuracy, usefulness and efficacy, and organizational fit. After the two formative evaluation rounds, several changes have been made to the BizDevOps Method and the new version of the artifact along with an evaluation form has been sent to the experts as a part of the third and final summative evaluation round. Overall, the BizDevOps Method was experienced as having the solid potential to provide the expected utility/benefit to organizations, with several further limitations and implications remaining. The BizDevOps Method was found to be flexible enough to accommodate to different organizations and helps them by visualizing, communicating and guiding the BizDevOps transition. These were the main identified strengths. The main limitations of the artifact were identified in the artifact's lacking coverage of the strategic and tactical level of the organizations as well as that the BizDevOps Method is challenging to implement for less mature organizations. As already noted, taking into account the boundaries of this thesis paper, these suggestions make for a solid base for further research to explore, something that is elaborated in the next chapter.

8. DISCUSSION

Chapter 8 discusses the practical implications of this thesis paper's research results. Second, the chapter addresses the main theoretical and practical research contributions to the research field of Agile and (Biz)DevOps. Third, research validity and reliability, along with possible limitations are covered. Finally, chapter 8 presents related work and also makes several suggestions for further research.

8.1 Implications

The objective of this research study was to design a holistic and comprehensive BizDevOps Method that helps IT organizations align DevOps and business goals to enhance enterprise agility and deliver high-quality, customer-centric software. Based on these desirable outcomes and the results of this research study, there are several implications for organizations considering the adoption of the BizDevOps Method. The research results have demonstrated that business requirements vary depending on the industry-specific context and the business models. Therefore, organizations should consider how to customize and accommodate the BizDevOps Method to their processes and organizational culture, so that they can leverage BizDevOps to maximize value for the client.

8.1.1 BizDevOps implications concerning people and organizational change

In terms of people, the research results imply the importance of organizational change and the importance of change management in supporting the formation of cross-functional, T-shaped teams consisting of Biz, Dev, and Ops roles. Henceforth, organizations geared towards BizDevOps need to contemplate how to promote constant communication and interactions among the teams in order to align them around the same goal. This implies the stimulation and establishment of continuous feedback and alignment loops (information flow) that enable the BizDevOps team to unify around end-user centricity. Moreover, this research paper has shown that it is imperative to embed systemic thinking and empower BizDevOps teams to take full ownership, thereby indicating culture and sharing as key performance enablers. This indicates that the organizations should focus a good portion of their efforts on building a strong cultural foundation for BizDevOps and work towards maturity through measuring and improving the teams' satisfaction, business understanding, and ownership perception.

8.1.2 BizDevOps implications concerning process and practices

Regarding the processes and practices of organizations heading towards BizDevOps, the research result imply that organizations should contemplate how to gear the entire process towards the end-user centric mindset. Therefore, careful thought should be given to establishing an iterative approach to define requirements, develop the software, and validate the business value delivered. This study furthermore suggests that organizations should look to establish and clearly pinpoint qualitative and technical feedback cycles on multiple levels throughout the process to feed into the continuous improvement and alignment with the end-user needs. That also means that companies need to apply measure-to-improve principles that are guided by leading and lagging indicators, value focus, and outside-in KPI prioritization. As a result, companies should seek to build resilience towards quality issues and responsiveness to changing end-user needs, driven by constant learning and improvement. In addition, as noted by the practitioners, problems across organizations often vary in nature and scale. As such,

further implications for organizations would be to identify what are the most occurring problems of their clients and to attach a set of problem-specific KPIs that solve targeted problems.

8.2 Research contributions

8.2.1 Theoretical contributions

Literature in the BizDevOps field was shown to be limited and is still strongly lacking empirical research. This thesis paper closed the empirical gap in the BizDevOps literature by conducting a qualitative study via interviews with experts in the field. Additionally, the thesis paper has set an example for future research by studying several IT organizations and their best practices vis-à-vis BizDevOps. This study is also one of the first to take a holistic and action-based approach that combines theoretical and empirical evidence to offer a solution for addressing the alignment of DevOps with business goals. Due to the empirical background of this research next to the literature review, this thesis provides insights into BizDevOps alignment practices connected to a more real-world context rather than insights based on idealization. Furthermore, this study is one of the first to have conducted validation research in the domain of BizDevOps literature. The BizDevOps Method was subjected to both formative and summative evaluation by field experts that were not seen in prior research of BizDevOps. It is for the above-mentioned reasons that this thesis paper significantly contributes to the (Biz)DevOps area of research.

8.2.2 Practical contributions

This thesis paper is the first known study to have designed a BizDevOps Method with a high generalization and abstraction ability which enables IT organizations to customize and align their processes with the BizDevOps Method. Designed based on literature insights and proven empirical practices, the BizDevOps Method guides organization in establishing customer-centric and continuous improvement mechanisms that bridge the gap between DevOps and business goals. More specifically, the method outlines the bridging steps for requirement specification, rapid feedback cycles, dynamic team formation, performance measuring and improvement among others. Furthermore, the designed artifact has been validated and acknowledged by industry experts, which implies the BizDevOps Method's potential in providing value to organizations in achieving a competitive edge through, faster time-to-market, operational excellence, and end-user experience.

8.3 Research validity, reliability, and limitations

This section of the thesis paper addresses the quality of this research as well as its limitations. More specifically, this matter concerns the validity and reliability of the research results and the overall research approach of this thesis paper, explained in chapter 3. The criteria used to evaluate the quality of research results have been adapted from Yin (2009): construct validity, internal validity, external validity, and reliability.

8.3.1 Construct validity

Construct validity refers to the extent to which the researcher establishes correct operational measures for the studied concepts. Considering that a part of this thesis paper includes research-based interviews and expert opinion, some collected data includes subjective judgment which may question construct

validity. Carter et al. (2014) suggest triangulation as a means to ensure construct validity in qualitative research. The authors define triangulation as a qualitative research strategy that uses multiple methods or data sources to test validity, develop convergence of information, and a comprehensive understanding of phenomena. This study applied method and data source triangulation by conducting a systematic multivocal literature review, a qualitative study, and focus group validation with experts. Data has been collected from a variety of sources, including academic literature, grey literature, expert interviews, expert feedback, and additional documents from the studied organizations. A systematic literature review has been used to minimize bias and maximize consistency in the selection of BizDevOps literature. Nevertheless, there is a chance that some concepts and related work has been overlooked or did not appear in the selected sample of papers due to the systematic way of reviewing. Also, to mitigate the biases in the qualitative study and validation part of the thesis, multiple organizations and people with diverse expertise have been included in the study. Namely, this study included BizDevOps experts from both the IT and business sides, coming from IT vendors, IT consultancies, and organizations with IT departments across different industries. Yet, a limitation in this regard is that most organizations studied have been represented by one or two experts.

8.3.2 Internal validity

Internal validity is concerned with whether a cause-and-effect relationship between a treatment and an outcome is trustworthy. The research draws several causal conclusions such as that a continuous customer-centric requirement engineering will lead to a better alignment of DevOps with business goals. As a result, internal validity becomes an important aspect of such statements. This thesis paper draws on pattern-matching and explanation-building suggested by Yin (2009) to ensure the internal validity of this research. Multiple IT organizations with DevOps teams were studied to identify common patterns in practices and processes for the alignment of IT and business goals. All identified aspects that were critical for managing the alignment of DevOps with business goals were explained in detail. What is more, open coding, axial coding, and selective coding were utilized to analyze the collected data and build relationships. As a result, this thesis paper has established both intra-case and inter-case data analysis. Therefore, it can be concluded that solid grounds have been built to prove the internal validity of this research.

8.3.3 External validity

External validity according to Yin (2009) is concerned with the general applicability of the obtained research results. In other words, it refers to the extent to which the findings can be used in different settings. To maximize external validity, this study relied on several (Biz)DevOps organizations and followed a replication logic. This means that the Business-DevOps alignment process was the global unit of analysis per company which was analyzed to see whether the findings were comparable. What is more, these findings were compared with the existing literature. Nevertheless, despite the aforementioned measures, the external validity of this thesis paper is limited to the number of companies analyzed in this study. Furthermore, there is a chance that some of the interview findings were based on views and experience specific to organizational culture and other factors. It is thus not feasible to claim the complete external validity of this thesis paper. Yet, with the measures taken, a sufficient degree of external validity can be argued.

8.3.4 Reliability

Reliability in the context of this research refers to the extent that the findings of the BizDevOps alignment method are reproducible irrespective of the researcher involved. Several measures have been taken to minimize the reliability threat. These measures include systematic and detailed documentation of results in each research stage. Protocols for conducting a systematic literature review (Appendix A), qualitative study interviews (Appendix C) as well as focus group session (Appendix E) have been developed and used for this purpose. Yet due to the strict confidentiality rules followed by this study, it was difficult to exactly describe the nature of each company, which poses a limitation to the reliability of this study. Nevertheless, having strived to identify patterns and common practices among companies, it is expected that repeated execution of this research will result in similar or the same practices in aligning DevOps with business goals.

8.3.5 Design science research quality

To demonstrate validity and reliability for the design science research part, this thesis paper uses the seven guidelines for design science research by Hevner et al. (2004). Based on these guidelines it can be concluded that all principles have been sufficiently fulfilled. Yet, guideline 3, design evaluation displays a limitation of this research and thus deserves to be addressed. Considering that the artifact validation has only been conducted with a limited number of industry experts in focus group sessions, the empirical validation of the BizDevOps Method, for example through experimentation could be conducted in future research. Nevertheless, all other principles and guidelines have been followed, which implies an overall valid and reliable design science research.

Table 8.1
Guidelines for evaluating design-science research

GUIDELINE	DESCRIPTION	APPLICATION
1. Design as an artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.	A BizDevOps Method is designed for the enhancement of enterprise agility through the alignment of DevOps with business goals.
2. Problem relevance	The objective of design science research is to develop a technology-based solution to important and relevant business problems.	Misalignment of DevOps with business goals has been identified as a relevant business problem. The designed BizDevOps Method is a technology- and organization-based artifact.
3. Design evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.	The BizDevOps Method was validated with industry experts through a focus group session. An empirical validation in practice through experiments and systematic observations remains.

4. Research contributions	Effective design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.	This thesis paper represents one of the first studies to explore the topic of BizDevOps from a holistic and solution-based perspective. This study is also one of the few empirical studies in the domain of BizDevOps.
5. Research rigor	Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.	A rigorous systematic multivocal literature review and an empirical qualitative study were conducted to construct the BizDevOps Method. The artifact was evaluated through expert opinions.
6. Design as a search process	The search for an effective artifact requires utilizing available means to research desired ends while satisfying laws in the problem environment.	The BizDevOps Method design process took an iterative approach to reach an optimal version of the artifact.
7. Communication of research	Design science research must be presented effectively both to technology-oriented as well as management-oriented audiences.	The research has been presented to a group of (Biz)DevOps professionals and IT managers. Also, this research was defended in front of Business Information Technology university researchers.

Note: Guidelines for evaluating design-science research. adapted from Hevner et al. (2004, p. 83)

8.4 Related & future work

8.4.1 Related work

A number of academic papers covering the topic of BizDevOps have been published after 15th May 2020, when the data collection from literature sources for this research was conducted. Therefore, all academic contribution claims made by this thesis paper refer to the period until 15th May 2020. Short exploratory research conducted on 19th November 2020 indicates three notable publications related to BizDevOps:

- Lohrasbinasab, I., Acharya, P. B., & Colomo-Palacios, R. (2020, July). BizDevOps: A Multivocal Literature Review. In *International Conference on Computational Science and Its Applications* (pp. 698-713). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-58817-5_50
- Sanjurjo, E., Pedreira, O., García, F., & Piattini, M. (2020, September). Measuring the Maturity of BizDevOps. In *International Conference on the Quality of Information and Communications Technology* (pp. 199-210). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-58793-2_16
- Sanjurjo, E., Pedreira, Ó., García, F., & Piattini, M. (2020, June). Process Reference Model for BizDevOps. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-6). IEEE. DOI: https://doi.org/10.23919/CISTI49556.2020.9141123
  (This paper is in Spanish and was selected based on the English title and abstract)

While the identified papers covered several similar topics and concepts such as continuous practices, BizDevOps maturity, and process description, none of the papers covered a holistic methodological approach in addressing the problem identified in this thesis paper; business-IT gap. Therefore, the relevance of this study still holds.

8.4.2 Future work

BizDevOps is a relatively young and unexplored field as mentioned before. While this thesis paper is the first to provide a BizDevOps Method to align DevOps with business goals and enhance enterprise agility, the space for future research to build further is substantial.

First, the validation method of this research consists of a limited number of experts and their opinion, which limits the validation intensity. In addition, the BizDevOps Method has not been applied in a real-world context which leaves many questions remaining. To further advance the knowledge about BizDevOps Method, its benefits, and applicability, this thesis recommends more thorough validation research. More specifically, technical action research (TAR) mentioned by Wieringa (2014) is a validation method that could be used to apply the BizDevOps Method in a real organization and learn about its effects in practice. This can potentially provide new insights for the refinement of the artifact.

Second, further studying of the BizDevOps Method raises the question of organizational change management and acceptance of the model. Therefore, it would be interesting to study BizDevOps in the context of organizational change to see how BizDevOps as a way of working fits with the existing frameworks in place and how it is being adopted by DevOps organizations. Furthermore, further research may focus on examining how feasible it is for BizDevOps to deal with the problem of legacy systems. What are the costs associated with the BizDevOps transition? Does BizDevOps provide enough flexibility to organizations to tailor their organizational structure and step away from outdated practices and legacy systems? What onboarding measures need to be taken and how to track BizDevOps adoption by the people in a certain organization? These are just a few example questions related to organizational change that further research could look into. Addressing these and similar questions might result in the identification of new principles that are needed for the successful adoption of the BizDevOps way of working.

Third, while this research scope primarily focuses on the BizDevOps Method on an operational level, several practitioners have indicated the need to examine the implications of BizDevOps for multi-level enterprise governance. Therefore, further research could focus on examining what are the critical elements and how would the BizDevOps Method change on tactical and strategic levels of organizations. This also implies the examination performance measuring and pinpointing KPIs that can guide targeted problem solutions on different scales.

Finally, another prominent topic discussed in DevOps is the security aspect which led to the development of DevSecOps among researchers. DevSecOps is an extension of the DevOps chain which strives to integrate security controls and processes into the DevOps software development cycle by promoting the collaboration of development, security, and operations teams (Myrbakken, 2017). Thus, the logical step further would be to add the security component to the BizDevOps Method. Since security is one of the integral elements for quality assurance it would be beneficial to explore how to develop a model that brings business, development, security, and operations units together under one roof, BizDevSecOps. Therefore, this thesis proposes future research in the direction of BizDevSecOps to examine how DevOps software development cannot just be geared towards business, but also security goals.

9. CONCLUSION

In conclusion, this thesis paper investigated the possibilities for IT organizations to align their business and IT domains with BizDevOps in order to enhance enterprise agility and deliver software according to business requirements. It has been noted that challenges in software development increasingly revolve around gaining a competitive edge in terms of operational excellence and software quality. It is therefore essential to close the business-IT gap by, breaking down the silo between the two, actively involving both domains in the process, and redistributing responsibility between IT professionals, who deliver reliable and stable IT systems and business professionals, who understand the rationale of IT systems from the business (end-user) perspective. As a result, the main aim of this thesis paper was to provide IT organizations with a robust BizDevOps Method that will allow them to achieve a high degree of operational excellence and end-user experience through a BizDevOps business-IT alignment. Operational excellence and maximized end-user experience imply that the organization needs to have a clear understanding of the business requirements, the agility to respond quickly to changing business needs, and the capability to align its people, processes, technology, and data towards continuously developing and delivering software with the end-user in mind.

9.1 Answering the research question

It is for this reason that the following main research question was developed:

How can a BizDevOps Method be designed to enhance enterprise agility within an IT organization?

To answer the main research question, this thesis paper followed a three-phased design science methodology of Wieringa (2014). For this purpose, a multivocal systematic literature review and 13 semi-structured expert interviews have been conducted. The gathered literature and practical insights have been used as input to design the BizDevOps Method. The results of this thesis paper have been validated with four interview respondents and nine additional experts that previously did not participate in the qualitative study. The main research question has been divided into four knowledge questions and one design question. Optimization of information flow through feedback cycles and business value measuring have been identified as crucial enablers of BizDevOps, which is why they are the main focus of the four knowledge question; two literature-based and two practice-based. The five sub-questions and the answer to them are as follows:

> **KQ1:** Which practices to optimize the information flow and shorten the feedback cycles throughout the whole software development process are mentioned in the literature?

The main findings derived from the literature, regarding information flow and feedback cycles, imply the need for cross-functional teams whereby the business stakeholders are active participants throughout the whole process. They are versatile individuals capable of translating user needs into business requirements, and continuously updating and managing the requirement backlog. Biz in DevOps are business value maximizers who are capable of bridging the end-user and DevOps, prioritizing tasks and resources to shorten and simplify feedback cycles that result in optimal service delivery. Section 4.6, 4.7, and 4.8 provided additional insights on how to optimize information flow within a BizDevOps software development context. BizDevOps teams should rely on additional continuous engineering practices such as continuous planning, requirements engineering, Lean principles, storytelling, and human-centered design among others. What is more, it is essential to embed quality improvement

mechanisms in the feedback cycles, through testing practices prescribed by Agile testing trifecta and shift-left testing, along with business process improvement.

> **KQ2:** Which practices to optimize the information flow and shorten the feedback cycles throughout the whole software development process are used in practice?

The literature and practical insights complement and support each other. Practical insights provided extended contribution across several aspects. The concept of T-shaped BizDevOps teams was additionally introduced, together with the elaborated role of Biz in bridging business and IT. Value stream mapping and hypothesis-driven business value definition with pre-defined success criteria have shown to be crucial value definition practices. Continuous planning and alignment principle, backlog management, WJSF, and BDD are requirement engineering practices additionally introduced by the practitioners. Ultimately, the practical findings indicate that the practices related to continuous feedback are wickedly structured, meaning that there are multiple qualitative and technical feedback cycles in different stages of the process, on different levels, with different characteristics, and in various settings. The key is to integrate and synchronize them all together in order to ensure frequent information flow and progress updates on delivering value in line with the set business goals. As such, technical and qualitative feedback loops need to serve as connecting corridors for the autonomous, T-shaped BizDevOps teams to maintain constant communication and end-user centric mindset during the entire process.

> **KQ3:** Which DevOps KPI metrics that communicate and capture business value are mentioned in the literature?

While no universal metric package has been identified, the obvious pattern in the literature indicates a symbiosis of result-oriented metrics across cultural, performance, and customer domains to measure the effectiveness of DevOps in capturing business value. Several highly influential performance metrics such as velocity and MTTR, as well as business-specific metrics such as revenue per customer, NPS, and ROI, have been identified. Additionally, a number of models for measuring KPIs towards business impact have been presented, which recommend the prioritization of value-focus outside-in metrics (primary, secondary and performance-enabling indicators). The literature findings also suggest the combined use of leading indicators to aid in business value and goal definition, and lagging indicators for reflecting on performance and value delivery.

> **KQ4:** Which DevOps KPI metrics that communicate and capture business value are used in practice?

The distinction between leading-lagging and external-internal indicators was also encountered during the qualitative study. Next to a standard set of indicators encountered in the literature such as lead time and velocity, the practical findings additionally contributed through the addition of user engagement metrics. How these indicators are specified on the unit level is something that has proven to be dependent on the organizational operating sector, business model, and maturity. Therefore, metrics are expected to vary across software product companies, IT service companies, and government for example. Also, the organizational maturity level has shown to be an influential factor in terms of indicator usage. More mature organizations demonstrated the capacity to combine leading/lagging and internal/external-facing indicators and channel the efforts towards hypothesis testing for correlations and value delivered. This approach was also reflected in these organizations' business value definition stage through structured value defining mechanisms that they established. On the contrary, organizations that were on a lower level of maturity demonstrated limited capacity in combining

leading/lagging and internal/external-facing indicators. Such organizations were mostly focused on internal and lagging indicators such as internal capabilities and basic financial indicators.

DQ5: How to design a BizDevOps Method?

The previously answered knowledge questions have served as input for the design of the BizDevOps Method. It has been found that in order for organizations to adopt BizDevOps a predominately intermediate or higher level of DevOps maturity is required. To construct a robust artifact such as the BizDevOps Method, multiple components, stakeholders, concepts, and practices from both business and IT domains needed to be integrated and synchronized together into a context-and-project-specific situational method that has a high level of abstraction and general applicability. The BizDevOps Method prescribes active commitment of both Biz and IT roles in cross-functional BizDevOps teams that facilitate and drive exploration, development, and validation phases of software development and delivery. The artifact puts a central emphasis on iterative thinking, information flow, the presence of frequent feedback cycles, alignment loops, performance measuring, continuous improvement, and end-user centrism. As a result of such principles, the BizDevOps Method strives to guide organizations in achieving desirable outcomes that include enterprise agility, business-IT alignment, minimum product variability, and maximized user experience.

9.2 Main contribution

The final section of this thesis paper briefly summarizes the main contributions of the entire research project below:

- First, through a thorough, multivocal literature review on BizDevOps, this thesis paper was able to identify the main gaps in the research and indicate the lack of empirical work on the topic.
- Second, this thesis paper is also one of the first to take a holistic and action-based approach that combines theoretical and empirical evidence to offer a solution for addressing the alignment gap of DevOps with business goals.
- Third, this thesis paper is the first known study to have designed a BizDevOps Method with a high generalization and abstraction ability that guides organizations in business-IT alignment and enhancing enterprise agility to deliver high-quality software and maximize end-user experience. This is therefore the main contribution of this work.
- Fourth, this thesis paper is one of the first to have conducted validation research in the domain of BizDevOps literature. The BizDevOps Method was subjected to both formative and summative evaluation by field experts that were not seen in prior research of BizDevOps.
- Finally, this thesis paper laid the ground and provided suggestions for further research such as BizDevOps in relation to further validation research, change management, multi-level enterprise governance, and risk management.

REFERENCE LIST

Alturki, A., Gable, G. G., & Bandara, W. (2013). The design science research roadmap: in progress evaluation. *PACIS 2013 proceedings*. URL: https://eprints.qut.edu.au/61626/3/61626.pdf

Baškarada, S., Nguyen, V., & Koronios, A. (2018). Architecting microservices: practical opportunities and challenges. *Journal of Computer Information Systems*, 1-9. DOI: https://doi.org/10.1080/08874417.2018.1520056

Chasioti, K. (2019). *BizDevOps: A process model for the Alignment of DevOps with Business Goals* (Master's thesis). https://dspace.library.uu.nl/handle/1874/383196

Carter, N., Bryant-Lukosius, D., Dicenso, A., Blythe, J., & Neville, A. J. (2014). The Use of Triangulation in Qualitative Research. *Oncology Nursing Forum*, *41*(5), 545-547. DOI:10.1188/14.onf.545-547

Drews, P., Schirmer, I., Horlach, B., & Tekaat, C. (2017, October). Bimodal enterprise architecture management: The emergence of a New EAM function for a BizDevOps-based fast IT. In *2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW)* (pp. 57-64). IEEE. https://doi.org/10.1109/EDOCW.2017.18

Elberzhager, F., Holl, K., Karn, B., & Immich, T. (2017, November). Rapid lean UX development through user feedback revelation. In *International Conference on Product-Focused Software Process Improvement* (pp. 535-542). Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-69926-4_43

Elliot, S. (2014). DevOps and the cost of downtime: Fortune 1000 best practice metrics quantified. *International Data Corporation (IDC)*.

Fitzgerald, B., & Stol, K. J. (2014, June). Continuous software engineering and beyond: trends and challenges. In *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering* (pp. 1-9). DOI

Fitzgerald, B., & Stol, K. J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, *123*, 176-189. http://dx.doi.org/10.1016/j.jss.2015.06.063

Forbrig, P., & Herczeg, M. (2015, September). Managing the agile process of human-centred design and software development. In *INTERACT* (pp. 223-232). URL: https://pdfs.semanticscholar.org/2446/8a5c8e911219430b916c4f760bd5ae794b73.pdf

Forbrig, P. (2017). Does Continuous Requirements Engineering need Continuous Software Engineering?. In *REFSQ Workshops*. https://pdfs.semanticscholar.org/171b/473beb1fb38643a72b6da8827d8eb6e736e8.pdf?_ga=2.22657592.227905876.1591886083-105399991.1589814924

Forbrig, P. (2018, April). BizDevOps and the Role of S-BPM. In *Proceedings of the 10th International Conference on Subject-Oriented Business Process Management* (pp. 1-8). https://doi.org/10.1145/3178248.3178250

Forbrig, P. (2018a). Use Cases, User Stories and BizDevOps. In *REFSQ Workshops*. URL: https://pdfs.semanticscholar.org/5a92/a773017cea26b422121cdc58953eca7e082b.pdf

Forbrig, P., & Dittmar, A. (2018, September). Integrating HCD into BizDevOps by Using the Subject-Oriented Approach. In *International Conference on Human-Centred Software Engineering* (pp. 327-334). Springer, Cham. https://doi.org/10.1007/978-3-030-05909-5_21

Forsgren, N. Humble, J., & Kim, G. (2018). *Accelerate: The science of lean software and devops: Building and scaling high performing technology organizations*. IT Revolution.

França, B. B. N., Jeronimo, H., & Travassos, G. H. (2016, September). Characterizing DevOps by hearing multiple voices. In *Proceedings of the 30th Brazilian Symposium on Software Engineering* (pp. 53-62). DOI: https://doi.org/10.1145/2973839.2973845

Garousi, V., Felderer, M., & Mäntylä, M. V. (2016, June). The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In *Proceedings of the 20th international conference on evaluation and assessment in software engineering* (pp. 1-6). https://doi.org/10.1145/2915970.2916008

Goldkuhl, G., Lind, M., & Seigerroth, U. (1998). Method integration: the need for a learning perspective. *IEE Proceedings-Software*, *145*(4), 113-118. DOI: https://doi.org/10.1049/ip-sen:19982197

Gruhn, V., & Schäfer, C. (2015, September). BizDevOps: because DevOps is not the end of the story. In *International Conference on Intelligent Software Methodologies, Tools, and Techniques* (pp. 388-398). Springer, Cham. https://doi.org/10.1007/978-3-319-22689-7_30

Henderson-Sellers, B., & Ralyté, J. (2010). Situational method engineering: state-of-the-art review. *Journal of Universal Computer Science*. URL: http://www.jucs.org/jucs_16_3/situational_method_engineering_state/jucs_16_03_0424_0478_henderson.pdf

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105. DOI: 10.2307/25148625. URL: https://www.jstor.org/stable/25148625

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

Kaiser, A. K. (2018). Introduction to DevOps. In *Reinventing ITIL® in the Age of DevOps* (pp. 1-35). Apress, Berkeley, CA. ISBN: 978-1-4842-3976-6. DOI: https://doi.org/10.1007/978-1-4842-3976-6_1

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*(2004), 1-26.

Korpivaara, I. (2020). Performance measurement in Agile development organizations. URL: http://urn.fi/URN:NBN:fi:jyu-202003302564

Kupiainen, E., Mäntylä, M. V., & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies. *Information and Software Technology*, *62*, 143-163. http://dx.doi.org/10.1016/j.infsof.2015.02.005.

Levi9. (2016). Continuous Delivery Whitepaper. Retrieved from Levi9, IT operations website: https://www.levi9.com/wp-content/uploads/2018/04/Our-view-on-Continuous-Delivery-Whitepaper-LeviNine-2016.pdf.

LogiGear. (2019, October 29). Continuous Delivery: Everything You Need to Know: LogiGear Blog. Retrieved June 14, 2020, from https://www.logigear.com/blog/continuous-delivery-devops/continuous-delivery-everything-you-need-to-know/

Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, *157*, 110384. DOI: https://doi.org/10.1016/j.jss.2019.07.083

Lwakatare, L. E. (2017). DevOps adoption and implementation in software development practice. *University of Oulu*. URL: http://jultika.oulu.fi/files/isbn9789526217116.pdf

Lwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., ... & Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Information and Software Technology*, *114*, 217-230. DOI: https://doi.org/10.1016/j.infsof.2019.06.010

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems*, *15*(4), 251-266. DOI: https://doi.org/10.1016/0167-9236(94)00041-2

Moreira, C. G., & de França, B. B. N. (2019). Towards a healthier collaboration at the business-development interface. *XXII Ibero-American Conference on Software Engineering, CIbSE 2019*. https://www.researchgate.net/publication/332684271_Towards_a_healthier_collaboration_at_the_business-development_interface

Myrbakken, H., & Colomo-Palacios, R. (2017, October). DevSecOps: a multivocal literature review. In *International Conference on Software Process Improvement and Capability Determination* (pp. 17-29). Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-67383-7_2.

Oates, B. J. (2005). *Researching information systems and computing*. Sage. ISBN: 9781446235447

Posta, C. (2015). Blue-green Deployments, A/B Testing, and Canary Releases. Retrieved October 10, 2020, from https://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/

Prat, N., Comyn-Wattiau, I., & Akoka, J. (2014, June). Artifact Evaluation in Information Systems Design-Science Research-a Holistic View. In *PACIS* (p. 23). URL: https://www.semanticscholar.org/paper/Artifact-Evaluation-in-Information-Systems-Research-Prat-Comyn-Wattiau/a432d62f815be0f4428b6cb5e58811b2a72ece7d?p2df

Ravichandran, A., Taylor, K., & Waterhouse, P. (2016). *DevOps for Digital Leaders*. CA. ISBN: 978-1-4842-1842-6. DOI: https://doi.org/10.1007/978-1-4842-1842-6

Rodríguez, P., Mäntylä, M., Oivo, M., Lwakatare, L. E., Seppänen, P., & Kuvaja, P. (2019). Advances in using agile and lean processes for software development. In *Advances in Computers* (Vol. 113, pp. 135-224). Elsevier. DOI: https://doi.org/10.1016/bs.adcom.2018.03.014

Rolland, C. (1998, June). A comprehensive view of process engineering. In *International Conference on Advanced Information Systems Engineering* (pp. 1-24). Springer, Berlin, Heidelberg. ISBN: 978-3-540-69434-2. DOI: https://doi.org/10.1007/BFb0054216.

Rouhani, B. D., Mahrin, M. N. R., Nikpay, F., Ahmad, R. B., & Nikfard, P. (2015). A systematic literature review on Enterprise Architecture Implementation Methodologies. *information and Software Technology*, *62*, 1-20. doi.org/10.1016/j.infsof.2015.01.012.

SAFe. (2020, February 13). Advanced Topic - Behavior-Driven Development. Retrieved from https://www.scaledagileframework.com/behavior-driven-development/#:~:text=Behavior-Driven Development (BDD),part of, specifying system behavior.

SAFe. (2020, February 13). Advanced Topic – Test-Driven Development. Retrieved from https://www.scaledagileframework.com/test-driven-development/

Satyal, S., Weber, I., Paik, H. Y., Di Ciccio, C., & Mendling, J. (2019). Business process improvement with the AB-BPM methodology. *Information Systems*, *84*, 283-298. DOI: https://doi.org/10.1016/j.is.2018.06.007

Schrader, I., & Droegehorn, O. (2018). Transforming Business Moments into Business Models. In *Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE)* (pp. 87-93). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp). https://csce.ucmss.com/cr/books/2018/LFS/CSREA2018/EEE3650.pdf

Seale, C. (Ed.). (2004). *Researching society and culture*. Sage. ISBN: 1526423081

Smeds, J., Nybom, K., & Porres, I. (2015, May). DevOps: a definition and perceived adoption impediments. In *International Conference on Agile Software Development* (pp. 166-177). Springer, Cham. ISBN: 978-3-319-18612-2. DOI: https://doi.org/10.1007/978-3-319-18612-2_14

Solajić, D., Petrović, A. (2019). Devops and Modern Software Delivery. Paper presented at Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research. DOI: https://doi.org/10.15308/Sinteza-2019-360-368

Top, Ö., & Demirors, O. (2019). Application of a software agility assessment model–AgilityMod in the field. *Computer Standards & Interfaces*, *62*, 1-16. ISSN: 0920-5489. DOI: https://doi.org/10.1016/j.csi.2018.07.002.

Tremblay, M. C., Hevner, A. R., and Berndt, D. J. (2010). "Focus groups for artifact refinement and evaluation in design research." In: Cais 26, p. 27. URL: https://doi.org/10.17705/1CAIS.02627

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a framework for evaluation in design science research. *European journal of information systems*, *25*(1), 77-89. DOI: https://doi.org/10.1057/ejis.2014.36.

Venkatraman, N., Henderson, J. C., & Oldach, S. (1993). Continuous strategic alignment: Exploiting information technology capabilities for competitive success. *European Management Journal*, *11*(2), 139-149. DOI: https://doi.org/10.1016/0263-2373(93)90037-I

Verschuren, P. J. M., & Doorewaard, H. (2015). *Het ontwerpen van een onderzoek [Designing a research]*. Amsterdam: Boom Lemma uitgevers.

Wiedemann, A., Wiesche, M., Gewald, H., & Krcmar, H. (2019, January). Implementing the Planning Process within DevOps Teams to Achieve Continuous Innovation. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*. ISBN: 978-0-9981331-2-6.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Wohlin, C. (2014, May). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (pp. 1-10). DOI: http://dx.doi.org/10.1145/2601248.2601268

Yin, R. K. (2009). Case study research: Design and methods 4th edition. In *United States: Library of Congress Cataloguing-in-Publication Data*. ISBN: 9781412960991.

Younas, M., Jawawi, D. N., Ghani, I., Fries, T., & Kazmi, R. (2018). Agile development in the cloud computing environment: A systematic review. *Information and Software Technology*, *103*, 142-158. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2018.06.014

APPENDICES

A

Systematic literature review protocol

Research goals:

- Summarize the existing base of knowledge
- Identify any gaps in the current research to suggest areas for further exploration.
- Provide good background/context that appropriately positions new research activities

Research strategy/process:

Concentrate on searching through scientific databases and collect the (x) most relevant outcome publications. Furthermore, all collected papers go through determined inclusion/exclusion criteria. The decision on whether to include or discard the papers will be taken through several stages. First round of screening will be done by reading through the titles. After that, the remaining papers' abstracts will be read to narrow down the selection further. Finally, the remaining papers will be read in full and to finalized the selection.

Scientific databases:
Scopus
IEEE Xplore
ACM Digital Library
AIS e-Library

Search engine for grey literature:
Google Scholar

Keywords:
BizDevOps | BizDev | DevOps
Enterprise agility
Business process
Alignment
Metrics

Search key: ("BizDev" OR "BizDevOps" OR "DevOps") AND ("enterprise agility" OR "business process" OR "alignment" OR "metrics")

Inclusion/exclusion criteria:

INCLUDE	EXCLUDE
Papers about software engineering in English	Papers that only conceptualize Agile methods
Papers about (Biz)DevOps and IT alignment	Papers that only conceptualize DevOps
Papers about (Biz)DevOps and enterprise agility	Papers that only cover (Biz)DevOps automation tools
Papers about (Biz)DevOps and business processes	Papers outside if the topic and not in English
Papers about (Biz)DevOps and business KPI's.	

Academic databases search results

Databases: Scopus, IEEE Xplore, ACM Digital Library and AIS e-Library
Search key: *(“BizDev” OR “BizDevOps” OR “DevOps") AND ("enterprise agility" OR "business process" OR "alignment" OR “metrics”)*
Search date: 16.05.2020

SEARCH PROCESS	AMOUNT
Papers found	210
After screening on title	42
After screening on abstract	26
After screening on full text	10
Adding forward & backward referencing	13

Grey literature search results

Search engine: Google Scholar
Search key: *(“BizDev” OR “BizDevOps”) AND ("enterprise agility" OR "business process" OR "alignment" OR “metrics”)*
Timeframe: 2015 – now.
Search date: 16.05.2020

SEARCH PROCESS	AMOUNT
Papers found	142
After screening on title	7
After screening on abstract	6
After screening on full text	4
Adding forward & backward referencing	6

B

Systematic literature review results and paper classification

The papers selected for the systematic literature review are summarized in table B.1. A short description together with contributions and research method of each paper is presented. Also the papers have been classified according to types: conference papers (C), journal papers (J), books (B), thesis papers (T) and whitepapers (W).

Table B.1 – Selection and classification of papers for the systematic literature review

PAPER	TYPE	DESCRIPTION
Fitzgerald and Stol (2014)	C	Introduction of BizDev as an extension of DevOps and conceptualization through an umbrella concept of continuous software engineering. The findings are based on a literature review.
Fitzgerald and Stol (2017)	J	Continuity of the previous article with the extension of continuous engineering practices. The article furthermore proposes a conceptual framework of BizDev and BizDevOps. The findings are based on a literature review.
Forbrig (2017)	C	Follow up to on the previous two papers and proposes continuous requirements engineering to be added to the continuous engineering conceptual framework. The findings are based on a literature review.
Forbrig (2018)	C	Proposes continuous Subject-oriented BPM to be added to the continuous engineering conceptual framework. The findings are based on a literature review.
Forbrig and Dittmar (2018)	C	Proposes continuous Human-Centered Design to be added to the continuous engineering conceptual framework. The findings are based on a literature review.
Forbrig and Herczeg (2015)	C	Discusses and combines Human-Centered Design and Agile Development process models to enhance the usability aspect in agile software development. The model uses SCRUM terminology but can arguably be used in conjunction with other agile processes. The findings are based on literature research.
Elberzhager et al. (2017)	C	Proposes Lean UX approach for rapid collection of customer feedback and improvement of the product. The authors propose a solution-based app for automatic collection of user feedback during MVP testing. The findings are based on literature research and practical observations.
Wiedemann et al. (2019)	C	Elaborates on continuous planning mechanisms; scalability, security and quality which lead to continuous innovation. This adds up to the continuous engineering conceptual framework. The findings of the paper are based on exploratory case studies using semi-structured interviews.
Moreira and De Franca (2019)	C	Addresses the involvement of business stakeholders in a DevOps environment. The paper defines roles, responsibilities and practices of Biz within DevOps. The findings are based on a literature review and a qualitative study using a Grounded Theory approach.

Gruhn and Schäfer (2015)	C	Proposes a solution-based platform for actively involving business stakeholders in a DevOps environment, where business conducts low-code activities and IT safeguards the process. The findings are based on a literature review and a single case study in which the platform was implemented.
Schrader and Droegehorn (2018)	C	Addresses the organizational aspect of BizDevOps and elaborates further on the idea of cross-functional teams with the business unit as an active participant in the mix. The findings are based on a literature review.
Drews et. al (2017)	C	Identifies and explains different roles, responsibilities and interactions within BizDevOps cross-functional teams working on an e-commerce platform. The findings of the study are based on a single case study.
Satyal et al. (2019)	J	Proposes AB-BPM methodology for continuous testing and improvement of deployed products and processes. This in turn allows for faster customer feedback and builds resiliency towards rollbacks. The findings are based on the literature and a validation experiment of the model.
Chasioti (2019)	T	Provides a high-level BizDevOps process model of software delivery that stresses user-centricity and active involvement of the business in the process. The findings of the paper are based on a systematic literature and case studies.
Kupianen et al. (2015)	J	Thoroughly examination of high-influence KPI metrics in agile software development. The paper also suggests several business metrics to extend the shortcomings shortcomings of agile metrics. The findings of the study are based on a systematic literature review.
Forsgren et al. (2018)	B	Proposes four essential DevOps metrics for delivering business value and elaborates further on the importance of each metric proposed. The findings of the book are based on qualitative and quantitative studies, extensive literature research, case studies and large-scale survey.
Elliot (2014)	W	Provides and classifies a number of DevOps KPI metrics that capture business value. The KPI metrics are based on industry best practices across 20+ Fortune 1000 companies. The findings of the paper are based on a research survey.
Ravichandran et al. (2016)	B	Proposes four main dimensions and sample metrics to measure the effectiveness DevOps in capturing business value. Additionally addresses problematic metrics and outlines a criteria checklist for adopting metrics. The findings are based on extensive literature research, industry insights and practical experience of the authors.
Korpivaara, (2020)	T	Expands on previously suggested solutions with agile KPI framework that integrates several aspects such as stakeholders, value chain, dimensions and category prioritization among others. The findings are based on a thorough literature review and empirical research through interviews.

C

Interview protocol

My name is Nikola Stankovic, a masters student of Business Administration – Digital Business track at the University of Twente in the Netherlands. This research is a part of my graduation thesis that focuses on how BizDevOps can align DevOps with business goals to enhance enterprise agility. The main goal of my graduation thesis is to develop a comprehensive BizDevOps Method that guides IT organizations in improving their enterprise agility. With this interview, I would like to gain practical insights on what practices within DevOps are used and how the alignment with business goals is achieved. Moreover, I am interesting in how to optimize information flow and accelerate feedback cycles in DevOps so that customer-centricity in agile software development can be maximized. I also aim to gain insights into what KPI metrics and approaches are used for measuring DevOps business performance. This in return would provide mechanisms for continuous improvement.

All practical insights gained from this interview will be synthesized with literature findings to form a strong foundation for the development of the BizDevOps Method. The interview session is expected take between 30 and 60 minutes. During the session questions related to the following aspects will be asked: involvement of business stakeholders in DevOps, requirement specification process, feedback loops, KPI metrics, software development process in a DevOps environment, etc.

Confidentiality

The interview session will be recorder for transcribing purposes. The information gathered during this interview will only be used for scientific purposes with respect to confidentiality. All data collected regarding people, company and examples given during the interview session will remain confidential. If at any point you wish to gain insight into the interview transcript or summary, this will be provided. To preserve the anonymity of all mentioned entities during the interview session, this graduation thesis will assign unique codes to participants and organizations. Moreover, the interview recording will remain private and will not be distributed to any other third party. Once the graduation thesis is complete, the interview recordings will be permanently deleted and the findings will exclusively be used for this graduation thesis.

Voluntary participation

Participation in this interview is voluntary. Therefore, it is up to you to decide if you want to participate in this interview. In case you wish to participate, you are kindly asked to sign the consent form before the start of the interview. If at any point you feel uncomfortable to continue with the interview or before the completion of data analysis, it is your right to stop. All gathered data will either be returned to you or will be destroyed permanently.

Thank you for taking the time to participate in this interview and help me with this research.

If you have read and agree with the information stated above, please sign your consent below.

Participant	**Researcher**
Name:	Name:
Date:	Date:
Location:	Location:
Signature:	Signature:

This interview session will take 30-60 minutes. For analysis purposes, I would like to record this interview. All recorded information will remain confidential at all times. With your consent, may I start recording?

…*Recording starts*…

Introduction

1. Please introduce yourself. What is your expertise, experience and role in the organization?

2. What is DevOps according to you and why did your organization opt for this methodology?
    a) What are the core benefits that you have experienced since adopting DevOps?

3. Please describe activities and steps in DevOps needed to develop, release and operate software?

BizDevOps team composition

4. What DevOps team roles exist in your organization?
    a) Which business functions are typically represented in your DevOps team?

5. Which activities in the DevOps software development cycle would you argue are the most dependent on the business unit and related stakeholders?
    a) What is the role of business stakeholders in these DevOps activities?
    b) How does the business department communicating with the DevOps team
    c) What mechanisms enable increased involvement of business stakeholders in the project?

BizDevOps and Continuous requirement engineering

6. Which methods, tools and techniques are used to specify requirements from the client?
    a) Who is typically involved in this process and in what role?
    b) In what way are these requirements translated and delivered to DevOps team?
    c) Please describe this process with some examples?

7. Which continuous requirement engineering mechanisms are in place to manage the product backlog?
    a) How to you manage feature changes and updates during the software development process?
    b) How are requirements prioritized for each release cycle?
    c) What are the future practices to improve scoping and requirement specification process?

8. How is the customer perspective captured in the requirement specification process?
    a) How is customer involvement ensured during the process?

Feedback cycles in BizDevOps

9. Which methods, techniques and practices does your organization use to collect feedback and shorten the feedback cycles?
    a) How often does this occur?
    b) Who is involved and what are their roles?
    c) Can you provide some examples?

10. What techniques and methods do you use to test the product?

BizDevOps KPI metrics for measuring and validating business value

11. Which methods, techniques and activities are used to track and measure business performance in DevOps?

12. Which internal aspects/dimensions in your organization are measured?
    a) Could you provide examples of culture and sharing metrics?
    b) Could you provide examples of metrics for internal processes examples?

13. Which external metrics are used to measure and validate business value?
    a) Could you provide examples of customer-centric metrics?
    b) Could you provide examples of financial value metrics?

BizDevOps

14. Have you heard about BizDevOps? If so, what does BizDevOps mean to you?

15. How does your organization align DevOps with business goals?
    a) What roles are important to facilitate this IT-Business alignment?
    b) Which practices should be in place to facilitate IT-Business alignment?

16. Could you provide some examples of any existing challenges/gaps between DevOps and business objectives in your organization?
    a) What would be the benefits of successful alignment in this case?

Thank you for your time and the information provided.

…Recording ends…

Can you recommend someone who would be interested in participating in this research? Later on, I will organize a session to discuss and evaluate the results of this research. Would you be interested to join? In case you have any questions or would like to receive the end result of this research, feel free to contact me.

Synthesis of literature and practical findings

The following table represents the synthesis of chapter 4 and chapter 5 findings. Additionally, the artifact has been broken down in components and the origin of these components has been indicated and connected to practical and literature components.

ARTIFACT BREAKDOWN	LITERATURE COMPONENT	QUALITATIVE STUDY COMPONENT
BizDevOps outcomes	4.3 Defining BizDevOps	5.3 Understanding of BizDevOps outcomes
BizDevOps teams	4.4 BizDevOps cross-functional teams 4.5 The role of Biz in BizDevOps	5.4 BizDevOps teams 5.5 Biz in BizDevOps
Continuous feedback loops	4.6 Continuous in BizDevOps	5.9 Continuous feedback cycles in BizDevOps
KPI metrics	4.9 Capturing value with BizDevOps metrics	5.8 Measuring and validation of business value
I Exploration phase	4.8.2 BizDevOps process modeling	5.6 Business value definition
01 Explore	4.6 Continuous in BizDevOps	5.5 Biz in BizDevOps
02 Define	4.9.2 Frameworks for BizDevOps KPI measuring	5.6 Business value definition
03 Refine	4.7 Subject-oriented approach in BizDevOps	5.7.3 BDD and use cases
04 Plan	4.6 Continuous in BizDevOps	5.7.1 Continuous planning and alignment loops 5.7.2 Backlog management and definition
II Development phase	4.8 BizDevOps and continuous BPM improvement	5.7 Requirement engineering
05 Build	SAFe	5.7.4 Scrum XP
06 Test	4.8.1 BizDevOps product and process testing, SAFe	5.5 Biz in BizDevOps
07 Deliver	SAFe	5.9 Continuous feedback cycles in BizDevOps
08 Operate	4.8.1 BizDevOps product and process testing	5.8.1 Continuous improvement
III Validation phase	4.8.1 BizDevOps product and process testing	5.8 Measuring and validation of business value
09 Capture	4.9 Capturing value with BizDevOps metrics	5.8.4 Internal/external indicators
10 Validate	4.9.2 Frameworks for BizDevOps KPI measuring	5.8.2 Validation through hypothesis testing 5.8.3 Leading and lagging indicators
11 Learn	4.6 Continuous in BizDevOps	5.8.1 Continuous improvement
12 Adapt	4.8.1 BizDevOps product and process testing	5.8.1 Continuous improvement

Focus group session protocol

My name is Nikola Stankovic, a masters student of Business Administration – Digital Business track at the University of Twente in the Netherlands. This research is a part of my graduation thesis that focuses on how BizDevOps can align DevOps with business goals to enhance enterprise agility. I have designed a BizDevOps Method based on the insights collected from the scientific literature and from DevOps practitioners working at agile organizations. The BizDevOps Method provides guidelines for agile organizations to align DevOps with business goals and improve their enterprise agility. Today, I am going to present the BizDevOps Method and would like to collect your feedback for the sake of improving the artifact. All necessary materials to familiarize with the designed BizDevOps such as project description, presentation slides and questionnaires will be provided to you in advance.

This focus group session is expected to take between 60 minutes, and will be recorder for analysis purposes. During this session, I will use an interactive presentation to present the artifact and discuss questions related to components of the designed artifact, the BizDevOps Method. The questions have been formulated based on the following evaluation criteria**:** understandability, completeness & accuracy, usefulness & efficacy and organizational fit. All questions that will be discussed in this session are found on the next page of this document.

Confidentiality

The information gathered during this session will only be used for scientific purposes with respect to confidentiality. All data collected regarding people, company and examples given during the session will remain confidential. If at any point you wish to gain insight into the session summary, this will be provided. To preserve the anonymity of all mentioned entities during the focus group session, this graduation thesis will assign unique codes to participants and organizations. Moreover, the recording will remain private and will not be distributed to any other third party. Once the graduation thesis is complete, the recordings will be permanently deleted and the findings will exclusively be used for this graduation thesis.

Voluntary participation

Participation in this focus group session is voluntary. Therefore, it is up to you to decide if you want to participate. In case you wish to participate, you are kindly asked to sign the consent form before the start of the session. If at any point you feel uncomfortable to continue with the session or before the completion of data analysis, it is your right to stop. All gathered data will either be returned to you or will be destroyed permanently.

Thank you for taking the time to participate in this focus group session and help me with this research

If you have read and agree with the information stated above, please sign your consent below.

Participant	**Researcher**
Name:	Name:
Date:	Date:
Location:	Location:
Signature:	Signature:

Evaluation criteria and questions

1. Understandability

- Do you understand the model?
- Are the given names, concepts and visuals straightforward and easy to understand?
- What could be improved on this regard?

2. Completeness & accuracy

- Do you agree with the identified steps?
- Is there anything missing
- What could be improved…Are any of the steps missing or redundant?

3. Usefulness & efficacy

- To what extent does the model address the intendent problem, the lack of customer-focus, enterprise agility and business-IT (dis)alignment?
- Do you think that BizDevOps Method could add value to your organizations and in general? Why/why not?
- What could be improved on this regard in the model?

4. Fit with the organization

- To what extent is BizDevOps applicable to your organization?
- What role is organizational maturity play a role in this case?
- To what extent does the BizDevOps team in the model reflect reality…is it feasible in reality?

5. General opinion and open discussion

- If you could summarize your opinion/takeaway of the BizDevOps Method in one sentence what would you say?