UTRECHT UNIVERSITY

MASTER THESIS

BizDevOps: A process model for the Alignment of DevOps with Business Goals

Author:
Kleopatra CHASIOTI

Supervisors:
Dr. Marcela RUIZ
Dr. Sietse OVERBEEK

Daily Supervisor:
Chris MANADIS

Utrecht University

accenture technology

Business Informatics
Department of Information and Computing Science

July 11, 2019

UTRECHT UNIVERSITY

Abstract

Faculty of Natural Sciences
Department of Information and Computing Science

Msc. Business Informatics

BizDevOps: A process model for the Alignment of DevOps with Business Goals

by Kleopatra CHASIOTI

Over the past years, software delivery speed and quality have been accelerated by the prominent adoption of agile and DevOps practices. This allowed software to provide higher value to customers at faster pace. Although, DevOps has managed to tear down the walls between the two teams, business has not always been involved in the whole process of software delivery. Additionally, DevOps teams include a Product Owner, who is representing business perspective and the question remains, if this is enough. The user requirements are translated in feedback loops through the PO who is usually struggling with a backlog with hundreds of requirements. The involvement of users and the higher management is very limited in the whole process. The goal of this research is to provide better alignment between business and DevOps in a new concept, named BizDevOps.

This research follows the Design Science methodology, based on Wieringa (2014). A Systematic mapping is performed for exploring the current body of knowledge and structuring the research area. The industry perspective is captured by means of semi-structured interviews with experts on the field, representing the business, software development and operations perspectives. As a result, a solution treatment has been proposed, to improve the current research problem. The BizDevOps process model has been designed, as the main contribution of this research. The artifact is validated with industry experts in an exploratory focus group session and several improvements are performed to the artifact.

The BizDevOps process model is a high-level representation of the software delivery process, that aims to extend the current DevOps process, in order to enhance user-centricity and multiple stakeholders alignment. In conclusion, BizDevOps is a new concept that aims to empower business to be more involved in the process of delivering software. It allows to interact with the DevOps team at various phases, at the start *Explore and Identify phase*. During *Development and Operations phase*, the business role is to ensure smooth software delivery. At last, a *Validation phase* ensures the delivery effort is satisfying the customers needs.

Keywords: DevOps, BizDevOps, Software Delivery, User-centered, Business Value

Acknowledgements

One more chapter comes to an end, even though it feels like yesterday, when I started the MBI. I feel extremely happy to have been through this journey of knowledge and growth. Despite the hard work, a major contribution to where I stand today comes from all the help and support, that I was lucky to have.

First to my list, I would like to thank my academic supervisors for their help and guidance throughout this process. Marcela, always being so enthusiastic and supportive and Sietse, for your valuable feedback and advice. A great appreciation goes to Chris, my daily supervisor, for helping me believe in this project and all your valuable professional advice.

I am grateful for the unconditional support of my family, that gave me the strength to follow my dreams and ambitions, miles away from home. My sweet mom, dad and brother, I owe all of this to you!

Last, but not least, a huge thank you goes to my friends. I have been extremely lucky to have you, lovely B, Amr and Oka, supporting me in hard times. Thanks for making me smile, offering your kind help, when I really needed it and for being the best buddies ever. A big thank you to my best friends from home, for always giving me motivation to continue. This journey wouldn't have been the same without you, friends!

Today, I feel more experienced and confident about my career and this is all thanks to you!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Agile methodology has become very popular among software development organizations, due to the advantage on addressing difficulties arising from dynamically changing requirements (Patanakul and Rufo-McCarron, 2018; Cleland-Huang, 2012). Agile principles give emphasis on human interaction, organizations being more responsive to change and fostering collaboration with the customers (Younas et al., 2018; Cleland-Huang, 2012).

While adopting an incremental and iterative development process, as in agile methods, helps organizations to meet constantly changing customer needs, DevOps comes into play, in order to accelerate software delivery speed (Krishna Kaiser, 2018). DevOps is here to bridge the gap between development and operations teams. By increasing collaboration and communication between these two teams and emphasizing automation via tools orchestration and performance monitoring throughout the software development lifecycle, efficient value delivery is ensured (Jabbari et al., 2016; Muñoz and Díaz, 2017). DevOps adoption finds a good support in organizations already adopting agile methods, as they both share several core values and principles (Nagarajan and Overbeek, 2018), such as Continuous Integration and Collaboration, Continuous Delivery (Benguria et al., 2018).

All the above mentioned benefits of Agile and DevOps adoption, have been facilitated by the advent of cloud computing technology, that provide both the infrastructure and resources needed for DevOps implementation (Farooqui, 2018; Di Nitto et al., 2016; Younas et al., 2018). Furthermore, the DevOps way of working is more and more promoted by other powerful technologies like microservices and containers (Farooqui, 2018), that have revolutionized the way of delivering software, by enabling software delivery in small and independent software increments (Balalaie et al., 2016; Farooqui, 2018).

As described, the scenery in today's software product development process is disrupted by emerging technologies and organizations must undertake actions to respond to these changes, by enhancing their agility (Abdelkebir et al., 2017). Therefore, Digital Transformation initiatives are crucial for organizations that want to adapt to current market dynamics. Organizations' motivation for Digital Transformation is to enhance user experience, optimize their operations and enable agility in their business processes for scalability and future growth (Shivakumar, 2018).

With all these benefits brought to Software Development, one thing is sure, DevOps is here to stay. Industry expertise is mostly prevalent in this domain and has contributed in providing knowledge on DevOps best practices for a smooth transition towards. As a result, breaking down silos between developers and operations teams has gained much attention and the cooperation between these two has been

pictured in a T-shaped team (Krishna Kaiser, 2018), a cross-functional team that is responsible from end to end for the software increment.

Nevertheless, several studies have identified the importance of integrating business capabilities into software development (Wiedemann et al., 2019; Gruhn and Schäfer, 2015; Blueprint, 2017; Fitzgerald and Stol, 2014; Schrader and Droegehorn, 2018; Wiedemann et al., 2019; Forbrig and Dittmar, 2019). One would wonder why this is needed, as long as the role of the Product Owner is in place, bridging the communication between the business stakeholder and the DevOps teams. The answer is that the market is pushing organizations towards aiming for operational excellence and customers are becoming more and more demanding, hence organizations should view the software development process as integral part of the their vision and strategy (Schrader and Droegehorn, 2018; Fitzgerald and Stol, 2014). Therefore, a closer collaboration between the people driving business value and those driving software development is necessary.

1.2 Problem statement

A close alignment between people, processes and the technology, is seen as a key factor for ensuring that business value is preserved as the product flows through the DevOps loop. Translating the business domain into software engineering domain comes with the challenge that the business management teams are only involved in presenting new requirements and reviewing the final software, but they do not actively participate throughout the development process (Gruhn and Schäfer, 2015; Blueprint, 2017). Therefore, the gap between people that drive the business value and software developers in the current DevOps process is becoming deeper (Wiedemann et al., 2019).

To address this gap, we explore the BizDevOps model, as an extension of DevOps, defined as the integration of domain experts with DevOps teams (Wiedemann et al., 2019). One main driver for an organization to adopt a BizDevOps model is to stay competitive and innovative, by providing higher customer satisfaction and higher quality software (Wiedemann et al., 2019).

The concept of BizDevOps is new to the scientific world. The term originates from industry needs to address the misalignment of business and development worlds. Our primary focus is to identify which mechanisms should be in place, in order to facilitate the Business, Development and Operations alignment. Therefore, a formalization of the BizDevOps model is necessary.

This research aims in leveraging a BizDevOps model, in order to continuously take new business and customer requirements into account. The main intention is to ensure an active participation of business stakeholders in the software development and operation process, in order to accelerate feedback mechanisms and at the same time ensure that the output of the development team is meeting the business goals and the end user expectations.

1.3 Research scope

This research is conducted in the context of developing quality software systems. The activities involved in the software delivery lifecycle are in the spotlight. Moreover, the DevOps process is studied, by emphasizing more the *people* and *process* dimension, rather than the *technology* one. This choice has been made due to the human-oriented nature of the alignment problem of this research.

In addition, the business involvement in the software delivery cycle is studied in isolation. This research is only concerned with the role of business stakeholders in identifying and capturing product requirements and planning for the software development lifecycle. Other aspects like, budgeting, portfolio management are considered out of scope.

Furthermore, the target group of this research is DevOps teams and their interaction with the business of the agile organization. This research adopts a view of organizations that are subject to agile and DevOps transformation journey and aim for process improvement. Agile organizations are on focus, due to their suitability in adopting a DevOps way of working. In addition, medium to large enterprises are studied, as in this context, the challenges of the alignment problem are more prevalent. Regarding the type of organizations, the research concerns Software Product Organizations, that have one or more DevOps teams for delivering software systems, as their primary operational product.

1.4 Research goals and objectives

For the purpose of this research, the high-level goal is to analyze the needs, goals, drivers, and requirements for software development in agile-DevOps context. Later, formalize a process model that uses agile principles for bridging the gap between business stakeholders and DevOps teams.

Driven by the challenges identified in the problem statement, this research explores the domain of BizDevOps in the scope of medium and large enterprises enterprises. The current state of the art provides support for bridging business and development, or development and operations, but it is missing a process or method that guides the application of BizDevOps in practice. For this reason, the following research objectives are defined:

1. Propose BizDevOps as a concept
2. Provide a process model for the alignment of business goals in the DevOps lifecycle

1.4.1 Research questions

The main research question of this study is:

> **"How to design a process model for the continuous alignment of business goals with DevOps?"**

With this question we intend to provide agile organizations with a reference model, that has a high abstraction level and highlights the most important mechanisms that facilitate such an alignment. To answer the main research question, the following knowledge (KQ) and design (DQ) questions are defined:

RQ1: What is the motivation of organizations to have continuous business alignment in their DevOps process? (KQ)

With this question, the needs and drivers that motivate organizations to seek for breaking down the silos between business stakeholders and DevOps teams are identified. The main intention is to acquire knowledge on Agile and DevOps process and the gaps and challenges these concepts address. Furthermore, the challenges of successfully scaling DevOps and agile practices across the whole organization

are explored, with the intention to identify the different alignment points between Business, Development and Operations.

To answer this research question, a Systematic mapping study is performed, that aims to structure the field of study and provide a theoretical baseline to designing the BizDevOps process model. Furthermore, expert interviews have been performed, in order to study the research problem in practice.

RQ2: How can we extend the DevOps process, in order to achieve continuous business alignment? (DQ)

This question is the core contribution of this research. In this question, the discontinuity present between the DevOps process and Business activities is addressed, by proposing a solution treatment, the BizDevOps process model. Due to being a design question, several changes are introduced to the DevOps processes, with the intention to extend on the existing process. For that, the knowledge acquired on the previous question is leveraged to evaluate on how to provide solution to the missing links. Furthermore, the artifact serves as a starting point for how organizations can apply these extensions in their teams.

This question is answered by conducting qualitative interviews with experts representing the business perspective and the software development and operation perspective.

RQ3: How is the BizDevOps process model evaluated by experts? (KQ)

After the design, validation of the artifact is important to evaluate the interaction of the artifact with the context. The result of this question is an assessment of the artifact designed in RQ2 and an improvement version of it.

To answer this question, the designed BizDevOps process model is evaluated by experts regarding completeness, efficacy in goal achievement, fit with the organization and fit with people.

1.5 Thesis outline

The remaining chapters of this thesis report elaborate on the undertaken process for the research execution. The outline and a short summary of the chapters is provided below.

Chapter 2: Research Approach describes the research approach followed in this study and elaborates on the techniques used to realize the research goal.

Chapter 3: Literature Review provides the scientific background of this research, by summarizing the results of the Literature Review. The Systematic mapping results are reported and the main concepts of this thesis are described.

Chapter 4: BizDevOps industry perspective presents the industry perspective of this research. The interviews results are reported and a combined perspective of scientific literature and industry is provided, which gives a complete picture of the *Problem Investigation* phase.

Chapter 5: Artifact Design elaborates on the *Treatment Design* process. The chapter provides the steps to design the artifact and a detailed explanation of the BizDevOps process model components.

Chapter 6: Artifact Validation describes the *Treatment Validation* phase. The focus group validation session results have been reported, which provide the rationale for improving upon the first design of the artifact.

Chapter 7: Discussion and limitations discusses the research results and contribution, along with the limitations of this research and the measures that were taken to mitigate the validity threats.

Chapter 8: Conclusions provides the answers to the research questions and proposes several directions for future research.

Chapter 2

Research Approach

This chapter provides an overview of the research methods and techniques utilized, to facilitate the execution of this research.

2.1 Research Framework

This research follows the Design Science methodology based on Wieringa (2014). The Design Science methodology has a focus on studying a designed artifact in its context and is well-suited for design problems in the field of Systems and Software Engineering, because of its solution-oriented nature. Adopting this methodology, provides us with the ability to analyze the BizDevOps design problem, by taking into account the social and knowledge contexts. Figure 2.1 highlights the main interactions between the object of study and the social context of the research, in order to assess the relevance of the designed artifact. In addition, a rigor cycle is performed, with the purpose of eliciting knowledge from the current state of the art contributions and provide own insights.

Three phases of the engineering cycle, namely **Problem Investigation, Treatment Design and Treatment Validation**, have been adopted, as a foundation for the activities performed in this research. The Design Cycle, as adapted to this research is shown in Figure 2.2. The next sections elaborates on all the activities, research methods and techniques that take place in each phase.

2.2 Problem Investigation

The goal of the Problem Investigation phase is to identify, describe and evaluate the research problem before a solution treatment is given, with the intention to improve or solve the problematic situation (Wieringa, 2014). In this phase, a better understanding of the research area is obtained. The research objectives and the corresponding research question have been defined. The involved stakeholders have been identified and their goals have been determined. A conceptual framework of the problem has been created, in order to structure the research.

A Systematic mapping study is performed, in order to determine the current state-of-the art in the literature. This research follows the approach as explained in Petersen et al. (2008). The Systematic mapping technique allows to perform a thematic categorization of the current literature and identify research gaps easier (Petersen et al., 2015). The thematic categorization analysis implies providing counts on the occurrence of a specific theme, in the sample of papers chosen (Petersen et al., 2008).

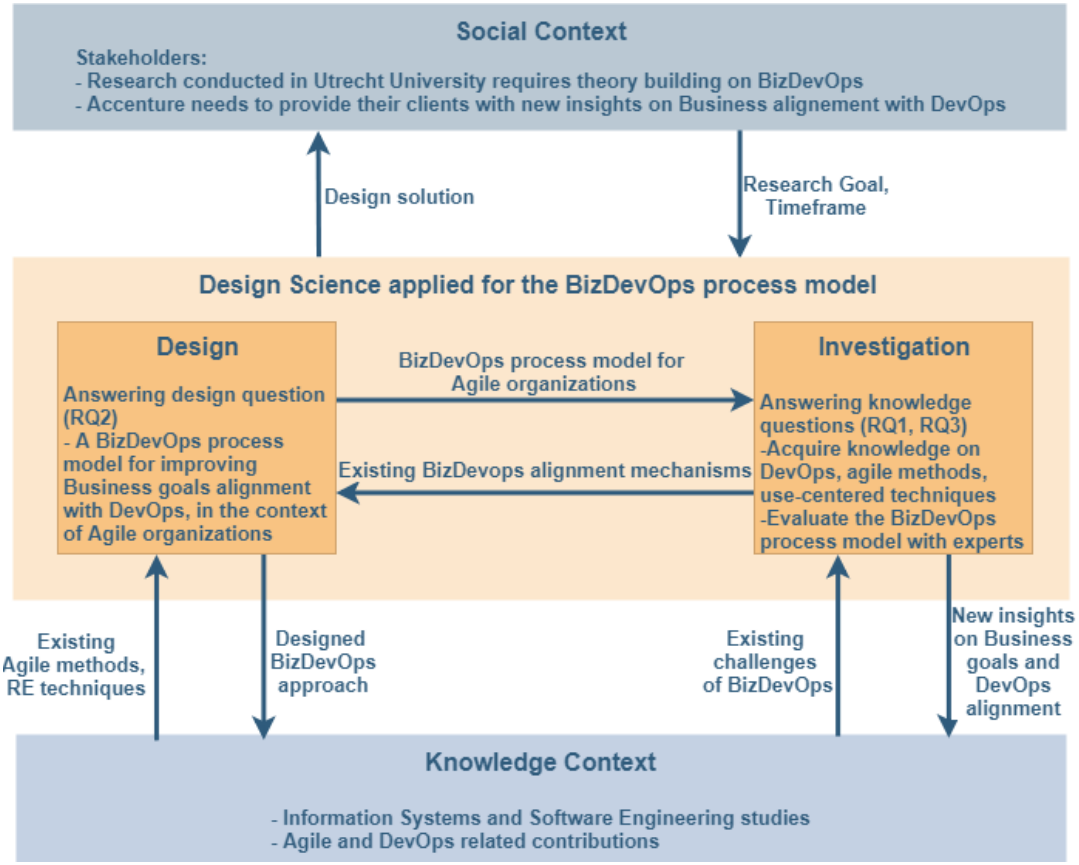

FIGURE 2.1: The Design Science framework by Wieringa (2014) applied in this research

2.2.1 Systematic mapping technique

A Systematic mapping study has been performed in this research, that facilitated the exploration of the current state of the art in literature. Due to the limited contributions existing on the concept of *BizDevOps*, a Systematic Mapping study has been chosen as a relevant way to provide an overview of the research area. The current body of knowledge is structured, by using a classification scheme and by analyzing current trends. The approach as described in Petersen et al. (2008) is adopted in this research. In order to keep a systematic and structured approach during throughout the process, the guidelines for performing a mapping study, as described in Petersen et al. (2015) are utilized. Figure 2.3 summarizes all the steps performed for the Systematic Mapping study, together with the outcomes of each step.

Planning

In the planning phase, a literature search protocol has been created, that can be found in Appendix A. At first, the goal of the Systematic mapping study has been defined, as follows:

1. Explore and summarize the current state of the art on the relationship between business and DevOps process alignment
2. Identify research gaps and trends in literature
3. Identify requirements for the artifact to be designed

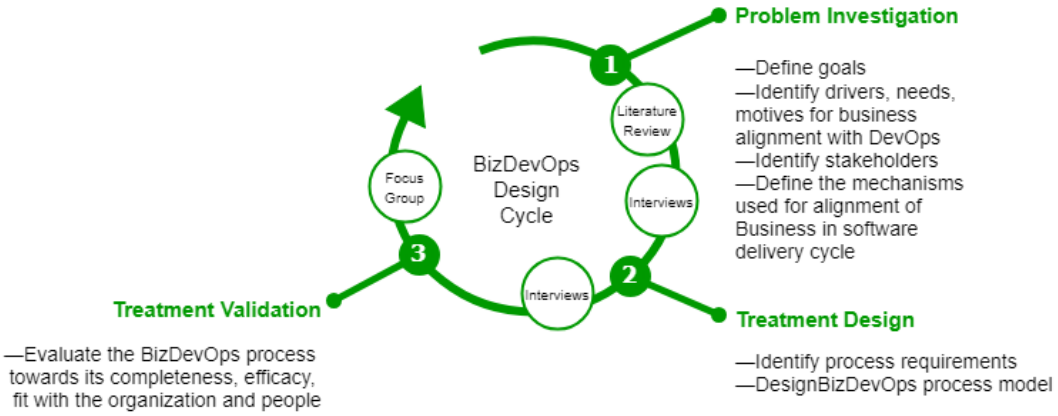

FIGURE 2.2: The Design Science cycle (Wieringa, 2014) applied to this research

A search strategy has been defined. Three databases has been selected for the search of four keywords.

Databases: ACM, Springer, Science Direct

Keywords: "DevOps" AND "business" OR "business process";
"agile" OR "process" AND "DevOps";
"traceability" AND "agile" OR "agile process";
"large scale DevOps

In addition, inclusion and exclusion criteria have been determined to achieve a representative selection of the search results, that can be found in Appendix A.

Executing

After performing the database search with the above mentioned keywords, the ten first search results have been documented in an excel file. This resulted to a total of 120 papers. After applying the Inclusion/Exclusion criteria, the sample size was reduced to 58 relevant papers. Later, a screening of abstracts has been performed and the papers have been classified regarding the ***a)*** *Research type* and ***b)*** *Contribution type* following the guidelines in the Systematic mapping protocol in Appendix A. As proposed in Petersen et al. (2008), Petersen et al. (2015), and Wohlin et al. (2013), the classification scheme of research papers in Requirements Engineering by Wieringa et al. (2005) is generally applicable and suitable to other research fields. Therefore, this classification scheme (Wieringa et al., 2005) has been adopted for the *Research type* classification. For classifying the publications related to the *Contribution type*, the definitions provided in Rodríguez et al. (2017) have been followed, adapted from Unterkalmsteiner et al. (2012) and Paternoster et al. (2014), that can be found in Appendix A.

At last, a thematic categorization (Braun and Clarke, 2006) using keywording technique on the sample of publications have been performed, in order to identify main concepts. These keywords have been used for creating a fourth classification scheme, *Area of Emphasis*, which relates to the concepts covered in the papers. This classification allows to group the sample publications in two categories, papers with an *Agile emphasis* and *DevOps emphasis*. This categorization scheme evolved in an

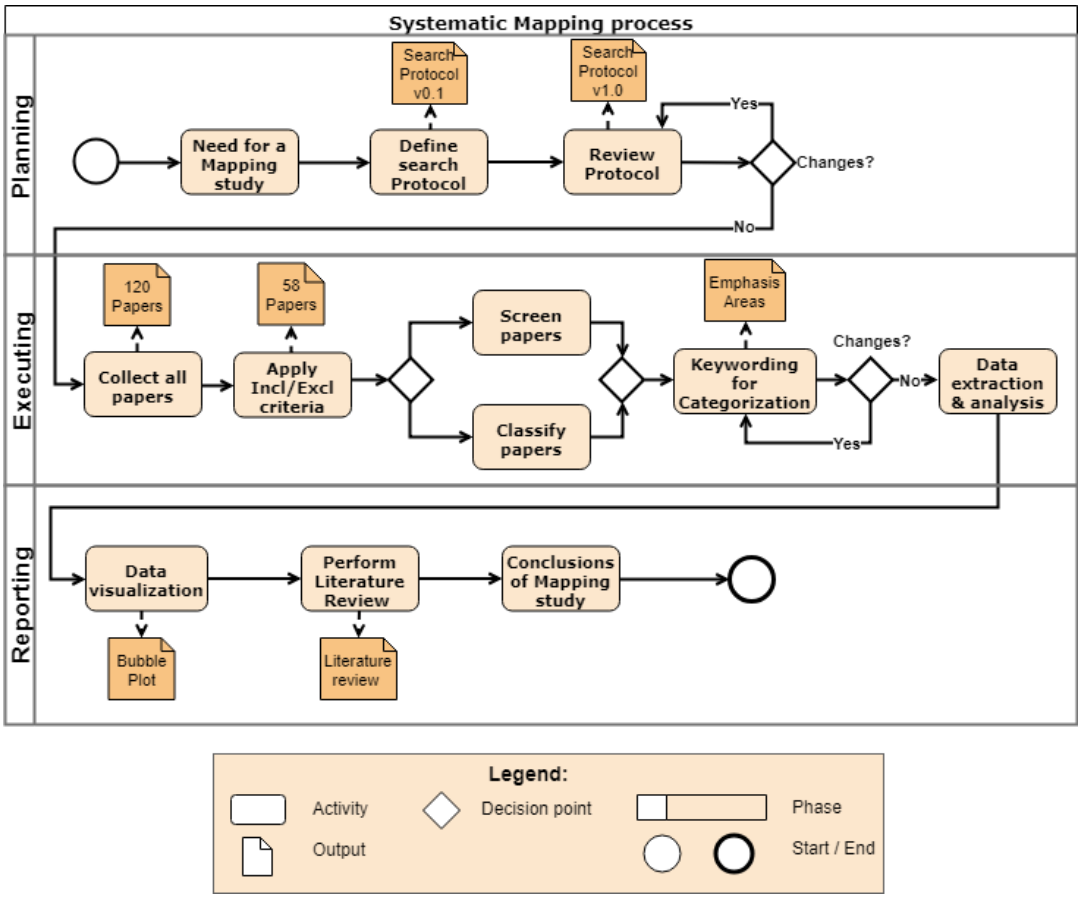

FIGURE 2.3: Steps followed for conducting the Systematic Mapping study

iterative manner, together with the researcher's understanding of the research area. The above *Area of Emphasis* categorization guides the analysis performed on the sample of publications. This activity is performed using the software *NVivo*.

Reporting

The last phase is reporting the results. The data analysis has been performed using *RStudio* and the results have been visualized in a bubble plot. The results form the Systematic Mapping are summarized and a further elaboration on the identified subcategories has been discussed in Chapter 3.

2.2.2 Interviews

Semi-structured interviews have been conducted for the Problem Investigation phase, in order to support answering the first research question (RQ1). An initial round of four interviews have been conducted with business and DevOps practitioners. Two interviews are conducted via personal interaction and two via Skype call. The interviews goal is to acquire a thorough understanding of how organizations deal with scaling Agile practices and how are DevOps practices seen in industry. An interview protocol has been created to guide the interviews, which can be found in Appendix B. Table 2.1 gives more details on the profile of the interviewees.

A second round of six interviews have been conducted, in order to acquire more detailed knowledge on the concepts identified on the first round of interviews. As

a result, another protocol has been created, as in Appendix C. Two interviews were conducted via Skype and the other four via personal interaction. Both rounds of interviews have been merged and analyzed by following the same approach. The insights have been jointly used for the Problem Investigation and Treatment Design phase of this research.

Interviewees sample: The interviewees sample was chosen out of convenience, considering the research timeframe. In order to acquire a representative overview of the research gap currently in practice, the interviewees were chosen out of these four generic categories: ***a)*** Service Delivery organizations (Internal): participants from Accenture, who provide the business and DevOps perspective from the consultancy view on different client organizations. The majority of the interviewees fall in this category, being the internal source of evidence for this research; ***b)*** Vendors (V): organizations that provide different solutions on tooling and infrastructure for DevOps adoption; ***c)*** Software Product companies (SP): agile organizations that have one or more internal DevOps teams and can reflect the actual demand or need for BizDevOps; ***d)*** Research community (R): since the term BizDevOps is very new in the scientific community, the perspective of researchers is also captured to compare with the industry needs; In Table 2.1, the column *Category* indicates in which of the four above mentioned Interviewee sample categories, the interviewee belongs to. In this thesis report, the interviewees are cited by using the below code, while the full reference can be found in Appendix D.

TABLE 2.1: Summary of interviewees profile

Code	Role	Years of experience	Category
iv1	PhD Researcher on BizDevOps	3 years	R
iv2	Lean Agile Coach	17 years	Internal
iv3	Project/Delivery Manager, Automation Architect	13 years	Internal
iv4	DevOps Lead	20+ years	Internal
iv5	DevOps Lead	5+ years	Internal
iv6	Manager, Scrum Master, PO, Transformation Consultant, SAFe	8 years	Internal
iv7	Operations Monitoring, User Experience	11 years	Internal
iv8	Agile DevOps Transformation Consultant, Coach, PO, Scrum master, Delivery Lead	12 years	Internal
iv9	Performance Engineer, Performance Analysis and Testing, Full-stack Monitoring	11 years	V
iv10	Innovation and Software Architect	20 years	SP

2.2.3 Additional Literature search

Next to the sources from the Systematic mapping, the snowballing technique has been used on the papers with direct contribution on BizDevOps, in order to acquire

a deeper comprehension of the new concepts, that arose from the first iteration of the *Problem Investigation* phase. The results are reported in Section 3.4.

2.3 Treatment Design

In this phase, the steps towards the creation of the solution treatment are described. The main artifact of this research is the BizDevOps process model, described in Chapter 5, that aims for the alignment of business and DevOps teams goals.

The process of analyzing all the findings follows the following rationale and is visualized in Figure 2.4. The results from the Systematic mapping, the two rounds of expert interviews and the extended literature search have been analyzed using the software NVivo. First, a decision on the level of abstraction for the solution treatment has been made, which resulted to the choice for representing the entire software delivery lifecycle. As a consequence, *traceability* related concepts were excluded. Next, the *challenges* to agile and DevOps adoption seen in practice, that were identified from the interviews have been combined with the literature review results, to generate a set of requirements for the treatment solution. The set of requirements were defined with the following criteria in mind: ***a)*** The requirements are in a high level of abstraction, thinking of the entire software delivery lifecycle as a series of activities performed by different actors (business stakeholders and DevOps teams) that communicate through multiple software engineering artifacts; ***b)*** The requirements tackle (most of) the *challenges* identified in the interviews;

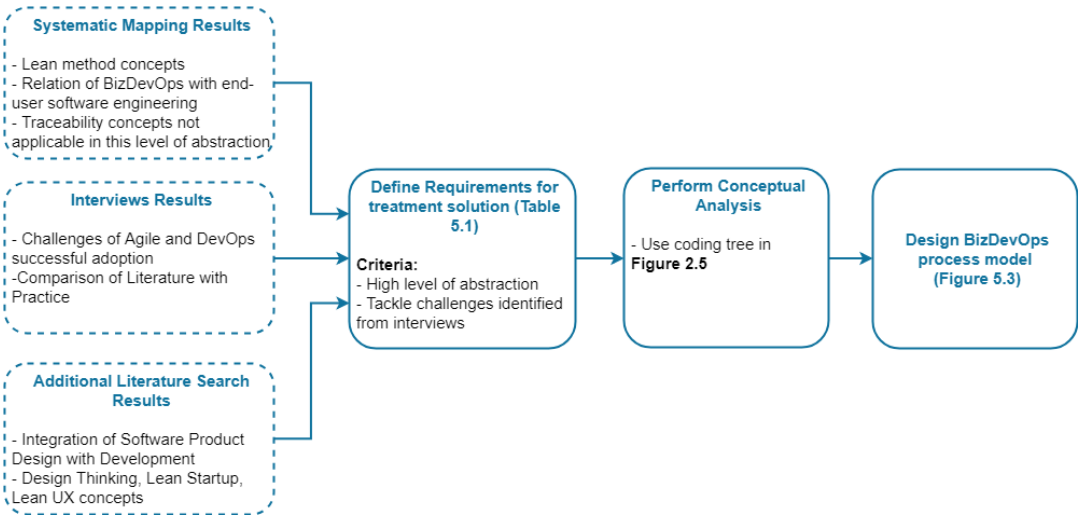

FIGURE 2.4: The process followed for data analysis and generation of the artifact

2.3.1 Conceptual Analysis

After the set of requirements for the treatment solution have been defined, the Conceptual Analysis has been performed to the interviews transcripts. Conceptual Analysis is a technique for analyzing and interpreting qualitative data. In this technique, symbolic data, such as interview transcripts are analyzed by searching for instances of the concepts in the conceptual model of the research at hand (Wieringa, 2014). Particularly, in this research, a coding tree has been define, which corresponds to the main concepts, that were needed to construct the treatment solution. The coding

tree nodes are namely *activities*, *artifacts*, *actors* and *technology*, as presented in Figure 2.5. Two scopes have been used to classify the above mentioned coding nodes, namely *Business* and *Development* scope, that represent the two domains of integration. Further explanation is provided in Section 5.2. The artifact has been designed by making use of the coded data and the additional literature findings.

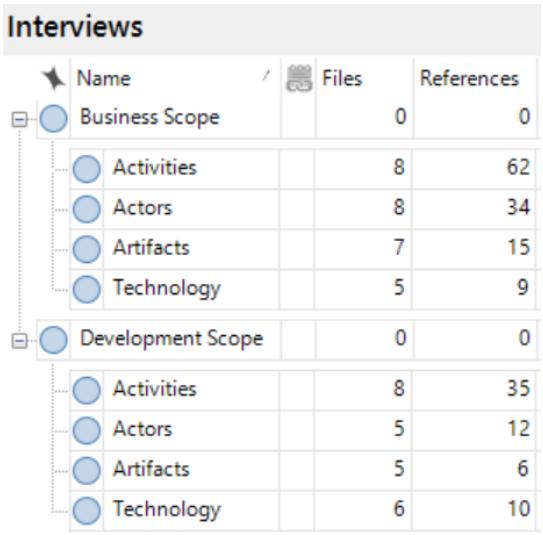

FIGURE 2.5: Coding tree used for the conceptual analysis of interviews transcripts

Methodological triangulation is a way to increase the validity of data interpretation, by using multiple independent methods for data collection (Wieringa, 2014) and is applied for the interpretation of literature, internal documents and interview data collected.

2.4 Treatment Validation

The Treatment Validation implies justifying whether the designed treatment contributes to satisfying the stakeholders goal, when applied in a real context (Wieringa, 2014). In this research, the designed treatment is validated with experts towards its completeness, realization of its goal and fit with the organization and people, in a focus group session.

2.4.1 Focus group

An exploratory focus group has been designed, following the design and execution steps, as described by (Tremblay et al., 2010). An exploratory focus group is used when the researcher aims to achieve rapid incremental improvements in the artifact design (Tremblay et al., 2010).

A validation model is created to simulate the interaction of the designed treatment with a real-world model. The participants of the focus group are experts in the field, hence the real-world simulation is perform by eliciting critical opinion from real-world stakeholders. Wieringa (2014) states that the experts are used as instruments to "observe" the artifact, by imagining the application of the artifact in a real case. In this research, the experts have evaluated the treatment based on imaginary industry scenarios, in order to assess how the proposed artifact is performing in these real-world scenarios.

Evaluation criteria

The evaluation criteria are selected from Prat et al. (2014). The BizDevOps process model is evaluated towards the dimensions and criteria presented in Table 2.2, as defined by (Prat et al., 2014).

TABLE 2.2: Validation session dimensions and evaluation criteria, adapted from Prat et al. (2014)

Dimension	Evaluation criteria	Description
Goal	Efficacy	The degree to which the artifact produces its desired effects.
Activity	Completeness	Accounts for the functionality of the artifact and whether is it missing any component.
Environment	Fit with the organization	Evaluates how the artifact would interact with the organization's context.
	Fit with people	Evaluates how the artifact is perceived by people.

Participants sample definition and recruitment

The focus group participants were chosen with regards to their expertise on the field. They should have experience with DevOps and software delivery. The desired sample size was 4 to 8 experts, to keep a small group for easier discussion moderation, given the focus group session duration time. The participants were all employees of Accenture and this selection has been made out of convenience.

The participants were approached with an invitation, one month prior to the session. An update on the session agenda has been sent two days before the focus group session. The goal of the session has been explicitly stated and the participants were informed that the session is recorded. Furthermore, their participation was voluntary and they had the right to leave the session at any time.

Participants profile

Seven participants attended the focus group session. Five of them were experts and actively participated in the discussion and two participants (participant6 and participant7) were mainly observers. A description of the participants profile is presented in Table 2.3. The full reference of the focus group participants can be found in Appendix D.

Focus group execution

In the focus group session, several means have been used to elicit the expert's opinion on the designed artifact.

- A power point presentation has been prepared to make the participants familiar with the designed artifact

TABLE 2.3: Focus group participants profile

Code	Role	Profile description
participant1	Developer	DevOps Experienced
participant2	Data Science Consultant	Experience with DevOps projects. More than nine years of experience.
participant3	Development Analyst	DevSecOps expert. One year of experience.
participant4	Development Consultant	Scrum Master Specialist, Agile Software Delivery, Business Intelligence. Nine years of experience.
participant5	Development Analyst	Enterprise Content Management. Two years of experience.
participant6	Data Governance	GDPR, Data Security, No previous DevOps experience.
participant7	Intern	No previous experience.

- Critical discussion on the designed artifact, after the presentation and an interactive post-it discussion is used
- Participants have been provided with a printed copy of the artifact and the validation session goal. They have been encouraged to draw and indicate changes on the model and write down any additional comments

Operationalization of the evaluation criteria

The focus group session was designed by keeping in mind the operationalization of the evaluation criteria. To measure the *Efficacy* and *Completeness*, an open discussion after the artifact presentation has been used. The participants have been asked to discuss about how the artifact can be applicable in a real-world context, by keeping in mind these two criteria and also leaving room for suggestions and improvements. The following questions have been asked:

1. Is the process model complete?
2. Do you understand the goal and purpose of the process model?
3. What can be improved?

To measure the *Fit with the organization* and *Fit with people*, two questions has been posed to the participants and they have been encouraged to write down their thought on post-its. The following questions have been used to stimulate the discussion:

1. Do you think that the BizDevOps process model can be adopted by organizations?
2. Is it feasible to work in BizDevOps teams? What should organizations do to adopt it?

The focus group validation session has been recorded and analyzed using the software NVivo. The audio recording has been coded, by using a predefined coding tree. The coding tree nodes correspond to the evaluation criteria, as shown in Figure 2.6. Along with the audio coding, the additional written evidence (post-its and comments on the printed model) have been analyzed.

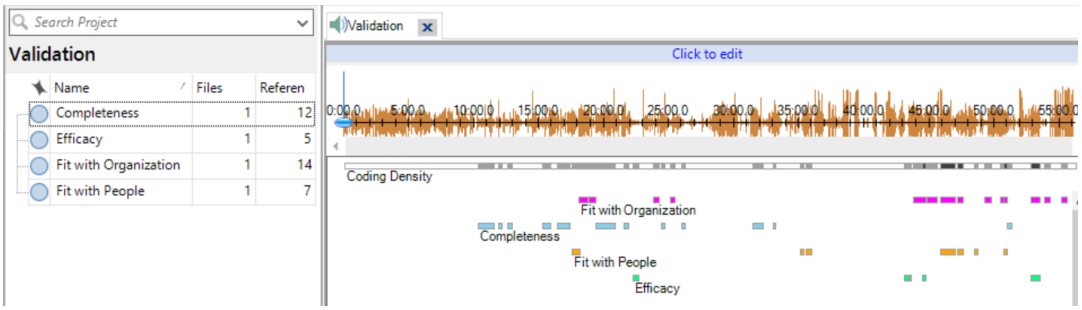

FIGURE 2.6: Coding tree used for the analysis of the focus group validation session

2.5 Chapter summary

This chapter gives an overview of the research approach followed in this study. The three phases of the Design Science cycle are elaborated in the context of this research and a description of the different utilized research techniques and methods has been provided. The different means of data analysis on the multiple sources of evidence are described and the systematic way of documentation is explained.

Chapter 3

Literature Review

In this chapter, the results of the Systematic mapping study are reported. The emerging themes have been used to structure the literature discussion section and a summary with the main findings is provided.

3.1 Systematic Mapping results

With the Systematic mapping, the current body of knowledge has been analyzed by means of both qualitative and quantitative techniques. A thematic categorization is performed on the sample of selected papers and two main areas of emphasis emerged. The *Agile emphasis* and the *DevOps emphasis* area.

The *Agile emphasis* area which entails contributions related to Agile methods. Three sub-categories have been used to classify the body of knowledge with respect to this area of emphasis, ***a)*** *Agile Adoption* ***b)*** *Agile Maturity* and ***c)*** *Traceability* use

Second is the *DevOps emphasis* area, where four sub-categories emerge: ***a)*** *DevOps Adoption* classifies papers providing the fundamental concepts and principles towards the DevOps journey ***b)*** *DevOps Maturity* sub-category entails contributions on DevOps maturity assessment ***c)*** *Continuous* *[1] sub-category summarizes concepts from Continuous Software Engineering practices. ***d)*** *Agile Business Process (BP)* sub–category entails contributions related to agile concepts in Business Process Management and the relation to DevOps.

The bubble plot presented in Figure 3.1 provides a visual representation of the current body of knowledge analyzed in this research. For the readers convenience, the bubble plot can be read with regards to the y-axis, which presents the *Area of emphasis*. The circles in the two axis intersections stand for the number of contributions, visualized by the size of the circle and the number in it.

On the top part of the bubble plot, the *Agile emphasis* area shows 10 contributions regarding the *Agile Adoption*. This study, found 3 frameworks or methods that contribute to guiding organizations towards Agile adoption and 4 publications contributing with lessons learned. Regarding *Agile Maturity*, this study found only 2 contributions evaluated in practice. Regarding *Traceability*, the majority of contribution (5 out of 10) use Validation research and they contribute with 4 models on attaining traceability and 3 contributions on lessons learned on traceability in agile projects. With regards to this mapping study, tools support is very limited for software artifacts traceability, as only 1 contribution has been identified.

On the bottom part of the plot, the contributions with *DevOps emphasis* are categorizes. Regarding DevOps Adoption, literature is rich in models, frameworks and methods. This mapping study identified also theory building contributions on

[1] Continuous * is an umbrella term used for a collection of Continuous Software Engineering practices (Fitzgerald and Stol, 2014; Fitzgerald and Stol, 2017)

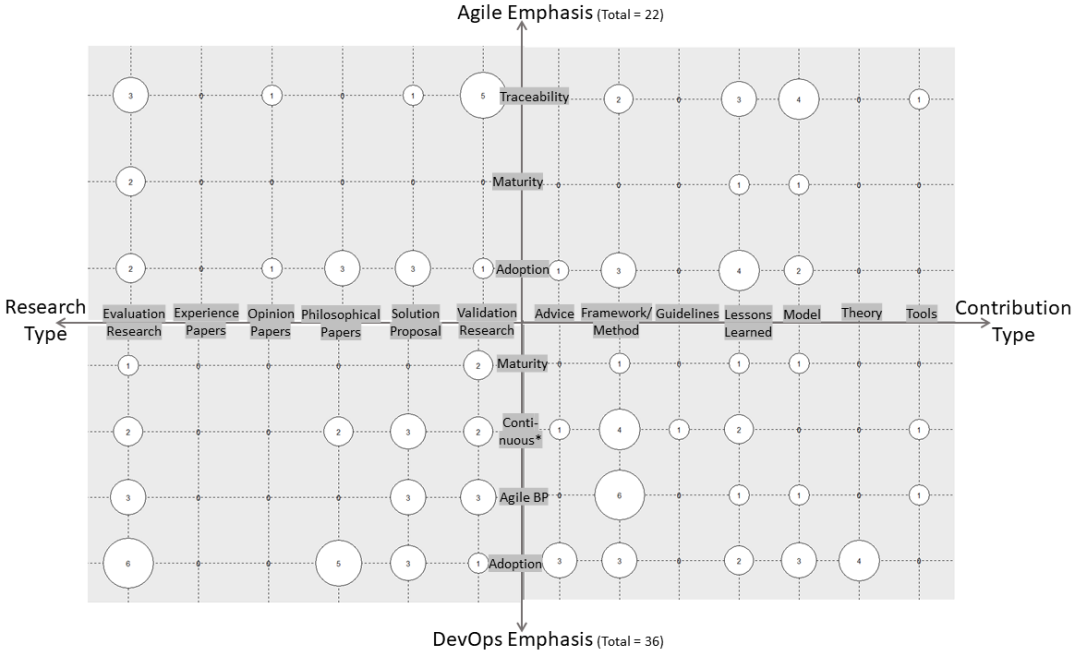

FIGURE 3.1: Systematic Mapping results visualized in a bubble plot

adopting DevOps practices. The majority (n=12) of the contributions in the *DevOps emphasis* area belong to Evaluation research, implies that the knowledge base is rich in practitioners perspective. Regarding *DevOps Maturity*, this study identified a few contributions (n=3). The *Continuous* * sub-category reveals 9 contributions emphasizing the use of Continuous Software Engineering practices in relation to DevOps. This study identified 4 frameworks and 2 lessons learned contributions in this area. In the *Agile BP* sub-category, 6 frameworks contribute to bringing agility in business process level, by using various techniques that are discussed below. These contributions are mostly Solution proposals or Validation research, for which a lack of practical insights through evaluation can be argued.

This mapping study contributes in structuring the research area like in Figure 3.1, with the purpose to identify gaps lack of scientific evidence. The following section elaborates further on the identified sub-categories, by highlighting important concepts.

3.2 Agile

3.2.1 Adoption

The Agile methodology represents a set of practices and principles, documented in the Agile Manifesto. These practices arose as a need for change from the traditional software development methodology, to a more lightweight development process (Cleland-Huang, 2012; Patanakul and Rufo-McCarron, 2018), that enables: the final product meeting customer needs (Top and Demirors, 2019); meeting deadlines and being conform to budgets (Top and Demirors, 2019); and rapid changes to products (Mergel et al., 2018). There are four core values of agile methodology, as described in Younas et al. (2018): *Individuals and interaction over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan*.

There are several widely used Agile methods. **EXtreme Programming (XP)** relies on continuous integration of small iterations of the project. Every iteration is an integrated part of code that is integrated and tested. This method addresses the problem of continuously changing requirements (Patanakul and Rufo-McCarron, 2018).

Scrum, relies on building tested software increments in short, fixed-length iterations of 2 to 4 weeks, called Sprints (Patanakul and Rufo-McCarron, 2018). Scrum emphasizes team self-management and feedback. A Scrum team consists of 5 to 9 developers. The work in the team is facilitated by the Scrum master. Scrum teams are designed to be self-organized and cross-functional, so that they remain flexible, creative and optimize productivity (**295**).

Kanban is an agile method that relies on three basic principles, that emphasize visualizing current workflow, limiting work in progress and managing lead time (Patanakul and Rufo-McCarron, 2018). Kanban allows for continuous changes in the team workflow and there are no prescribed roles, as in Scrum. Therefore, this method is suitable for environments with continuously changing task priorities.

Lean software development originates from the manufacturing industry and focuses on two principals: keep things simple by simplifying workflows and reduce waste, like removing unnecessary process steps, in order to avoid delays in software delivery (Krishna Kaiser, 2018).

Literature emphasizes that there is no "one way fits all" for Agile methods adoption on all organizations. This mapping study identified Agile adoption variations related to:

i) Organization size
Diebold et al. (2018) address the journey of small or medium-sized enterprises to adopting agile processes, through software process improvement (SPI). SPI initiatives serve as a means for companies to reach upon improvement goals, such as achieving shorter time-to-market, or increasing quality. The most affected areas of improvement after agile adoption were Project Planning and Assessment and Control and Quality testing.

The Scaled Agile Framework for Enterprise (SaFe)[2] is a framework developed to help organizations cope with scaling their agile practices. This framework is based upon agile and Lean principles and utilizes a DevOps maturity model that help organizations assess the stage of maturity (Top and Demirors, 2019; Patanakul and Rufo-McCarron, 2018).

ii) Sector or operational domain
Patanakul and Rufo-McCarron (2018) provides a framework for agile adoption in the public sector. The authors address the challenge of adapting an agile method of working in traditional business processes, that are based on long planning and milestones. Moreover, agile adoption comes with the need of standardization and regulatory compliance, which is imposed by regulatory authorities (Laukkarinen et al., 2018)

Nevertheless, it is crucial for the organizations that adopt agile methods to sustain and leverage these practises, in order to maximize their benefits on productivity, improved quality and faster time-to-market (Senapathi and Drury-Grogan, 2017). Senapathi and Drury-Grogan (2017) provide a model for sustained Agile usage. The authors identify three groups of factors (Agile Team, Technological and Organizational) that influence the effective and sustained use of Agile methods in organizations. The three groups of factors are summarized in Table 3.1 and additional related

[2]https://www.scaledagileframework.com/ Date accessed: January 2019

factors found in this mapping study are presented. Also, challenges related to these agile factors are presented in the last column.

TABLE 3.1: Agile Factors and challenges

Agile Factor	Related Factors	Challenges
Agile Team Factors	Self-organized teams (Patanakul and Rufo-McCarron, 2018; Jesse, 2018) Culture of sharing knowledge and experience (Jesse, 2018) Early customer engagement (Patanakul and Rufo-McCarron, 2018)	Culture change (Senapathi and Drury-Grogan, 2017; Top and Demirors, 2019) Misconception that documentation is not needed (Patanakul and Rufo-McCarron, 2018) Difficulty in communication due to distributed teams or locations (Benguria et al., 2018; Younas et al., 2018)
Technological Factors	Establish metrics measurement (Senapathi and Drury-Grogan, 2017) Implementing test Automation (Patanakul and Rufo-McCarron, 2018)	Integration with standard business processes and tools (Patanakul and Rufo-McCarron, 2018)
Organizational Factors	Top management support (Senapathi and Drury-Grogan, 2017) Dedicated role of Agile Coach (Senapathi and Drury-Grogan, 2017)	Change management (Patanakul and Rufo-McCarron, 2018) PO commitment (Patanakul and Rufo-McCarron, 2018)

Agility refers to the capability of an organization to respond efficiently to internal and environmental changes (Abdelkebir et al., 2017). Important external factors influencing agility are customer and market changes. Customer satisfaction is a business driver and Kuranuki et al. (2014) propose a business model for custom software development suitable for agile organizations. They emphasize the frequent customer involvement in the development process, in order to manage increased costs originating from changes in customer requirements. Another external factor is organization's journey towards digitization. The emergence of Digital Transformation requires organizations to become more responsive to change and embrace agile practices across the whole organization. This implies for a reorganization of roles in digital enterprises, aiming to spread agility in all enterprise levels. One example of such a change is mentioned by Jesse (2018), that imply a collaboration between Data Scientists or Chief Analytics Officers with Business Analyst in the business department.

3.2.2 Traceability

Software traceability is "*the ability to interrelate any uniquely identiable software engineering artifact to any other, maintain required links over time, and use the resulting network to answer questions of both the software product and its development process*", as defined

by the Center of Excellence for Software Traceability (CoEST) (Cleland-Huang et al., 2014, p. 849).

Requirements traceability has gained interest in Software Engineering. Due to the fact that requirements are the starting point in software development projects, any changes that may arise to them can influence the development process as a whole, resulting in necessary architectural changes, test cases generation changes and changes in validation activities with the user (Carniel and Pegoraro, 2018). Over the past years, research has contributed to promote traceability by developing sophisticated tools, promoting the establishment of strategic planning for traceability, automating the process of creating and maintaining trace links and developing new queries and visualization techniques (Cleland-Huang et al., 2014).

User feedback provides a rich source for new requirements, hence feedback - requirement traceability is an important trace link to obtain. These traceability links can be used for later impact analysis of implementing new features or requirements, or assess the importance of their implementation (Dzvonyar et al., 2016). Traceability matrices are used for keeping control over software artifacts trace links, by automatically generating and mapping relationships among multiple artifacts (Jeong et al., 2018).

Controversial ideas on traceability arise from the Agile perspective. Traceability is seen as a practice that doesn't bring any direct contribution to working software, therefore establishing and maintaining trace links is usually abandoned (Palmer, 2014). Traceability practices are seen as heavy-weighted, time-consuming and redundant from the Agile perspective (Cleland-Huang et al., 2014). From a Lean perspective, it is seen as a process that is producing waste (Palmer, 2014). Due to that, trace links in agile projects are characterized as inaccurate, conflicting and outdated. Therefore, traceability has been proven to be difficult to attain and maintain.

The benefits of applying traceability management in agile projects is presented are risk minimization, by being able to estimate the impact of changes; reduced development costs and increasing team productivity, by reducing rework and saving time; test coverage, by tracking requirements from inception to the whole delivery cycle, dependencies are easily identified and testing is facilitated; visibility in the development process(Carniel and Pegoraro, 2018).

3.2.3 Maturity

Agile maturity assessment models help organizations assess their agile capabilities, identify gaps and guide them towards undertaking the necessary steps for achieving the desired state (Top and Demirors, 2019). In this mapping study, the Software Agility Reference Model (AgilityMod) was identified. The model defines four levels of agility, namely Not implemented(level 0); Ad-Hoc(level 1), where fundamental development activities are implemented, but the team cannot fully leverage the benefits of agile. This should be a transition period to the second level; Lean(level 2) is characterized by two attributes *iterative* and *simple*. *Iterative* refers to the capability of obtaining frequent customer feedback and *simple* refers to eliminating non-value added processes; and Effective(level 3) the development team achieves technical excellence and a culture of learning.

In order to achieve and sustain agile benefits in an organization, measuring and monitoring is an important aspect. Senapathi and Drury-Grogan (2017) identifies the need for adopting metrics both from the *technical perspective*: like lead time, delivery time; and the *business perspective*: customer satisfaction, meeting market needs and expectations, in order to continuously monitor the project's agile performance.

3.3 DevOps

3.3.1 Adoption

DevOps has emerged as a paradigm that aims to tightly integrate developers with software operations team (Wettinger et al., 2016). While there is no single definition of DevOps, we adopt the following from Jabbari et al. (2016, p. 6) : *"DevOps is a development methodology aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices."* DevOps is often conceptualized as the intersection between Development, Operations and Quality Assurance (QA), which puts an emphasis on monitoring of the running software (Forbrig, 2018a).

The most widely perceived DevOps principles are represented with the acronym CALMS. *Culture* emphasizes the importance of human component for a successful adoption. DevOps relies on a culture of collaboration, responsibility and ownership, experimentation and innovation. *Automation* of tasks through the whole delivery pipeline is crucial for fast delivery. Following *Lean* principles for optimizing processes and being efficient. *Measurement* of metrics throughout the delivery pipeline enables monitoring and responding over measurable outcomes. *Sharing* emphasizes the elimination of silos and that speed and efficiency can be achieved when knowledge is shared between people (Krishna Kaiser, 2018).

Nagarajan and Overbeek (2018) conducted an evaluation research, where six drivers for Agile and DevOps adoption in large financial organizations are identified namely, ***a)*** Agility and Customer-centricity ***b)*** Efficient Value Delivery to Customers ***c)*** Cooperative Culture ***d)*** Empowered People ***e)*** Focus on Continuous Improvement ***f)*** Process and Stakeholder Alignment

Smeds et al. (2015) provide insights on DevOps capabilities and enablers, that are presented in Table 3.2. This can be a starting point to understanding the nature of journey towards DevOps adoption.

People

The merge of development and operations teams into one DevOps team, puts the people factor on focus, implying for a cultural change, in order to support the DevOps way of working. DevOps teams are characterized as cross-functional teams, responsible from end-to-end for the software increment (Krishna Kaiser, 2018). The cultural enables in Table 3.2 give the essential characteristics of a DevOps team.

A DevOps team typically consists of the following roles (Krishna Kaiser, 2018): Product owner (PO), from the business organization and is the owner of the product backlog; Scrum master, leads the development team; Developers, responsible for coding; Testers, involved in developing test scripts and executing functional and nonfunctional tests; and the Operations people like, Database or System administrator for database and configuration management; Service manager, responsible for managing services from the incident, problem, change, and other service management areas;

Except from the team reorganization and the cultural insights introduced to it, the organizational perspective is also emphasized. An organization adopting DevOps should have the managerial support (Jones et al., 2016). Especially in large-scale agile organizations, where the agile principles should be present in all three levels, team, project and portfolio level (Nagarajan and Overbeek, 2018).

TABLE 3.2: DevOps capabilities and enablers (Smeds et al., 2015)

Capabilities	Continuous planning Collaborative and continuous development Continuous integration and testing Continuous release and deployment Continuous infrastructure monitoring and optimization Continuous user behavior monitoring and feedback Service failure recovery without delay
Cultural Enablers	Shared goals, denition 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 Conguration management for code and infrastructure

Process

A typical DevOps process consists of the following activities, that represent the standard software delivery process : Plan, Code, Build, Test, Release, Deploy and Operate. In DevOps, the activities from Code to Deploy are automated. The automation is enabled by a series of integrated tools and the process is referred to as the DevOps pipeline. The DevOps capabilities in Table 3.2 provide the main processes that should be established in the DevOps pipeline.

Continuous Integration(CI) process enables integration of multiple developers commit and provides rapid feedback about code changes (Rahman et al., 2018).

Continuous Delivery (CD) enables the deployment of the integrated code into production (Krishna Kaiser, 2018; Shahin et al., 2017). The distinction between continuous Delivery and Continuous Deployment should be made explicit. In Continuous Delivery, every change in the code is proven to be deployable at any time. This implies that every code change should be deployable to environments before production (like User Acceptance Testing or Pre-production). The deployment to production is however, not a continuous activity, because a manual intervention should be made (Sharma, 2017). When the deployment to production activity is automated, than Continuous Deployment is performed. So, the ready to deploy, integrated increments form CI/CD are made available and usually, what is deployed to production is a set of these integrated parts, tha might typically form a feature, or a full application or service (Sharma, 2017). However, achieving Continuous Deployment requires a relatively DevOps mature situation, where CI and CD are up and running smoothly. Therefore, Continuous Deployment is *optional* for DevOps organizations.

One reason why Continuous Deployment is an option is that many organization

are required to adhere to policies that require a manual approval process before deployment to production. Another reason is the *segregation of duties*. It implies that the team that has to do the deployment is different than the team that contributes to the development of the deployable piece of code (Sharma, 2017).

It is also important that the DevOps team performs Continuous Monitoring of users interaction. By monitoring and measuring various technical and business metrics, feedback loops can be used for continuously improving software development processes and development of product features. These metrics are then interpreted and utilized for the following planning activity, which initiates the next DevOps increment (Babar et al., 2015).

Technology

One of DevOps principles is process automation. Technological factors are the enablers of automation in the DevOps toolchain. All the above mentioned continuous practices are facilitated by the orchestration of powerful tools and technologies that enable a smooth and automated workflow between all process steps (Farooqui, 2018; Krishna Kaiser, 2018).

Tool support is crucial for enabling process automation and orchestration. However, having the processes drive the need for tools implementation is more important than trying to automate without considering the processes and architecture (Krishna Kaiser, 2018).

3.3.2 Maturity

In order to strengthen the DevOps capabilities mentioned in Table 3.2, DevOps process improvement is necessary for organizations. The Capability Maturity Model Integration (CMMI) models have been applied to guide process improvement initiatives, however they are not a good fit for DevOps process improvement (Rong et al., 2016).

Feijter et al. (2018) provide a DevOps Competence model, summarizing all the necessary perspectives and focus areas that a DevOps organization should empower. They identify three perspectives: Culture and Collaboration; Product, Process and Quality; and Foundation, providing the architecture and infrastructural baseline. The DevOps Maturity Model they provide can help organizations assess the maturity of their DevOps capabilities and guide a fine-grained maturation. Furthermore, the model identifies both internal and external stakeholders involved in DevOps activities.

3.3.3 Continuous *

Continuous Integration, Continuous Delivery and Deployment are core Agile practices utilized in DevOps, aiming to foster Development and Operations alignment. Furthermore, Continuous Software Engineering practices are discussed in Fitzgerald and Stol (2014) and provide the foundations for bridging the gap between the three pillars, Business with Development and Operations. The term BizDev is used in literature to refer to the continuity between business strategy and software development (Fitzgerald and Stol, 2014; Dittrich et al., 2018).

BizDev originates from the Continuous Software Engineering (CSE) model by Fitzgerald and Stol (2014), as shown in Figure 3.2. CSE emerged from the concepts of Enterprise Agile and Beyond Budgeting, which claim that the benefits of agile

software development should also be supported by an agile way of working in related organizational functions such as finance and HR (Fitzgerald and Stol, 2014). Therefore, suggesting agile adoption as a cross-organizational capability. In Figure 3.2, the continuous practices in the development and operations domain follow the DevOps philosophy. The added practices of Continuous Innovation, Improvement and Planning are thought to bridge the gap between business strategy and Development (Gruhn and Schäfer, 2015; Forbrig, 2018a; Fitzgerald and Stol, 2014).

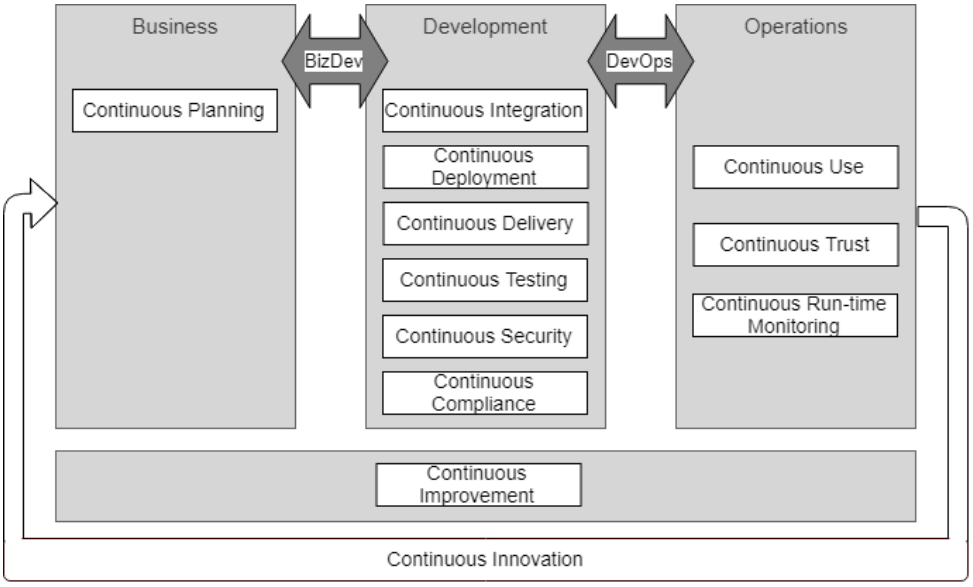

FIGURE 3.2: Continuous *: A holistic view on activities from Business, Development, Operations and Innovation (Fitzgerald and Stol, 2014, p. 5)

Continuous Planning is defined as, "*A holistic endeavor involving multiple stakeholders from business and software functions whereby plans are dynamic open-ended artifacts that evolve in response to changes in the business environment, and thus involve a tighter integration between planning and execution.*" (Fitzgerald and Stol, 2014, p. 4). Continuous Improvement is based on lean principles of data-driven decision-making and elimination of waste, which lead to small incremental quality improvements that can have dramatic benets and are hard for competitors to emulate (Fitzgerald and Stol, 2014). Continuous Innovation is embracing it all. It is defined as a sustainable process that is responsive to evolving market conditions and based on appropriate metrics across the entire lifecycle of planning, development and run-time operations (Fitzgerald and Stol, 2014).

According to Dittrich et al. (2018), there are three main challenges of Continuous Software Development: design and architecture to support CSE; quality and test automation; changing processes and management within and beyond the software organization.

BizDevOps

The term BizDevOps has emerged over the last years in literature. While there is no agreed definition of what the term implies, the different perspectives are discussed in this section.

A BizDevOps approach is discussed in Gruhn and Schäfer (2015), from the perspective of End User Software Engineering. The authors present a software platform

in the form of a framework, that allows the business department to actively participate in the software development. The BizDevOps approach by Gruhn and Schäfer (2015) addresses the boundary between IT and business departments and allows for: ***i)*** business people to express and review requirements in a hands on manner, thus reducing the necessity of knowledge transfer from IT to business department; ***ii)*** IT department to govern the application development process and ensure high quality of software artifacts; ***iii)*** an integrated and automated toolchain to enable the development pace. However, this approach is targeting a rather specific case of software systems development, the one of end-user approach.

Forbrig (2018a) extends this idea, by providing an approach on how Business Process Modeling enables BizDevOps. In a BizDevOps approach, BPM can give all stakeholder groups a common language and a reference point to aid in decision-making. Following the idea that business stakeholders should be more involved in the software development process, Forbrig (2018a) make use of S-BPM, a subject-oriented business process modeling approach that allows business people to specify processes and workflows and translate them into code, by means of domain specific language. This way domain experts can be actively involved in the development process. In the same line, later contributions Forbrig and Dittmar (2019), Forbrig (2016b), and Forbrig (2016a) are emphasizing the use of Continuous Requirements Engineering, Continuous Business Process Modeling and Human-Centered Design (HCD) in the agile development lifecycle, in order to extend and utilize the CSE practices by Fitzgerald and Stol (2014) and Fitzgerald and Stol (2017).

Forbrig (2018b) discusses the BizDevOps concepts from a Requirements Engineering perspective. Requirements are identified as the the starting point of the SDLC and that business people are the main source of these requirements. The author supports the idea that storytelling can serve as a means of communication between business and development. Storytelling is a technique used to support managers in communicating company's values, vision and culture in the the whole organization Forbrig (2018b) and in a BizDevOps approach, it can facilitate the collaboration and communication of management, business analysts and development.

BizDevOps is discussed by Wiedemann et al. (2019), with a focus on Continuous Planning process. The authors support the idea that organizations can achieve Continuous Innovation by establishing a sustained mechanism of Continuous Planning of customer requirements. This contribution adopts a team-level view of BizDevOps and identifies three mechanisms for Continuous Planing. A BizDevOps team should be able to achieve planning *Scalability*, which means they have responsibility over the service they deliver and must recognize and proactively make planning decision related to scaling the service. Second, *Security* in the planning activity empowers the BizDevOps team responsibility for accurate decisions. The third mechanism is *Quality* in the planning activity, which relates to the teams culture of collaboration and sense of ownership.

The importance of having a cross-functional BizDevOps is discussed by Schrader and Droegehorn (2018). The authors recognize the need for establishing new business models to empower the customer interaction. Figure 3.3 presents the authors envision of a BizDevOps team, that is bridging the gap between domain knowledge from business and Development activities.

3.3.4 Agile business processes

Business Process Management (BPM) refers to a collection of tools and methods for achieving an understanding of, managing, and improving an enterprises' process

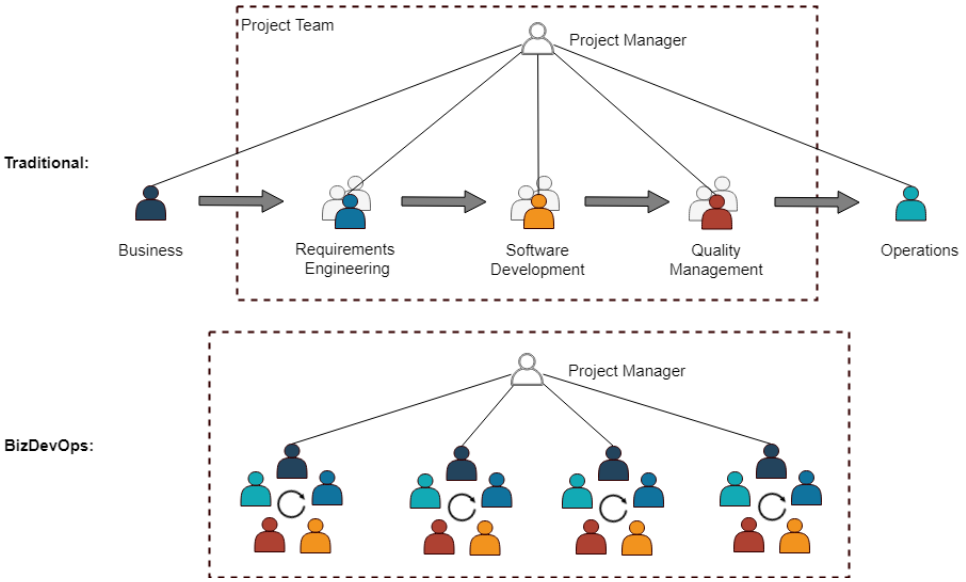

FIGURE 3.3: Visualization of the difference between the tradition and a BizDevOps team, adapted from Schrader and Droegehorn (2018, p. 91)

portfolio (Schulte et al., 2015). They are considered to be one of the most important assets for organizations, since an appropriate management of them helps companies to adapt their business goals and structures to environmental changes quickly, while maintaining or improving their competitiveness (Pérez-Castillo et al., 2019). The BPM lifecycle includes the following activities: process analysis, design, implementation, monitoring and improvement (Ståhl et al., 2016). Organizations rely on business processes (BPs) for their proper functioning, which may include operational, transactional, strategic or design processes (Babar et al., 2015). BPM roles are usually related to business roles, such as domain experts and analysts. But, it is important that technical roles, like developers are involved in BPM activities, as process models provide a good point of communication between these two types of stakeholders (Ståhl et al., 2016).

The concept of Agile BPM arose as a need for change in the traditional way of performing the BPM lifecycle. Organizations are embracing more and more the Agile way of working, therefore having static processes is no longer an option to cope with the dynamically changing requirements (Babar et al., 2015).

This mapping study identified the following BPM practices, which contribute to bringing agility in process management:

1. Using AB testing for BP deployment
   One of BPM goals is achieving incremental process improvement through process redesign activities (Satyal et al., 2017), that enable improved efficiency and effectiveness of the processes (Ståhl et al., 2016). A usual BPM redesign lifecycle starts with analyzing the As-Is process, designing the To-Be process, deploying the process in operation and monitor to evaluate the improvements. For monitoring the actual improvement of the redesigned process, the concept of AB testing from DevOps is used. This implies that, both the old and redesigned process are operating in parallel and validated in real time towards the same conditions (Satyal et al., 2017). Satyal et al. (2017) provide the AB BPM technique for facilitating the business process improvement process.

2. Process refactoring
   Quality assurance of BP models can be assessed by their *understandability* (the degree to which users recognize the product is appropriate for their needs) and *modifiability* (the degree to which a business process model is effectively and efficiently modied without introducing defects or degrading performance) (Pérez-Castillo et al., 2019). Not all software or system development activities start from process modeling, which later guides the software or system requirements to be developed. Reverse engineering is the process of analyzing a system to create a high-level representation of it (Pérez-Castillo et al., 2019). Business process model refactoring techniques are used to increase the quality of business process models. They intend to perform changes in the internal structure of the process model, without altering its external behaviour, resulting in improved understandability and modifiability.

3. Workflow customization strategies
   Workflows are used along with business process models, as they represents an automatically executable procedure of processes. Multi-tenant SaaS providers offer their application as a sigle run-time instance that is shared among multiple tenants. Due to the tenants different requirements, it is important that the SaaS application offers them a customized workflow. Therefore, Makki et al. (2018) emphasized that architects should choose for a workflow customization strategy as early as possible in the design process, in order to design quality software. System's scalability and support on DevOps activities is influenced by this early choice on workflow-driven multi-tenant SaaS applications.

3.4 Additional findings

This section summarizes the additional literature findings, that emerged after the Systematic mapping was conducted.

Literature on BizDevOps is supporting a human centered approach on DevOps (Forbrig, 2016c; Forbrig and Dittmar, 2019). The main intention is to emphasize that having good technical solutions is not enough. The involvement of the end user, through Continuous Software Engineering approaches and human-centered design is necessary, in order to enhance the customer value delivered (Forbrig, 2016c). Carell et al. (2018) identify the changes imposed in Requirements Engineering processes from organization's shift to DevOps and other emerging paradigms of Digital Transformation. The authors quote this change as a competence gap between business people focusing only on the business perspective of requirements and software developers focusing on the technical side of the software. Requirements and usability engineering provide the interface of stakeholders communication with software development and this is a key point for improved collaboration of the two disciplines.

User Centered Design

Agile Software Development methods focus on measuring value by the usefulness of the software product delivered to the customer, while very little focus is placed on usability of the product. Usability is not a primary concern in Software Engineering and is usually categorized as non-functional requirement and quality attribute of the system (Brhel et al., 2015). In User-centered design (UCD), a good alignment of end-user goals and needs is ensured throughout the product development. The field of user-centered design, human-centered design, and usability engineering are

driven by a continuous end-user evaluation of the end product and use an iterative development approach. In general, all the user-centered approaches follow these generic steps: *research*, *specification*, *design* and *evaluation*.

Due to these similarities, literature has gained much interest in combining Agile development and user-centered design, in order to maximize the value delivered to the end-user. This provides our motivation to embed a human dimension in the BizDevOps approach.

Another popular user-centered method is **Lean UX**, which combines three methodologies: Design Thinking, Agile development and Lean Start-up (Elberzhager et al., 2017).

1. Design Thinking: principles of Design Thinking support that design methods can be used in every phase of the project, even by non-designers, as this technique encourages collaboration across different roles. It has a user-centered approach and also combines the business view and the technological feasibility of the idea.

2. Agile development: Lean UX inherits the agile values of incremental and iterative development, with a great focus on feedback, that allows for rapid changes, strong collaboration and communication culture and customer involvement.

3. Lean Start-up: this techniques is based on the creation of a Minimum Viable Product (MVP), by means of continuously applying the "build-measure-learn" loop. This way, the team can minimize project risks and support quick feedback by means of rapid prototyping.

Often, the use of user centered approaches is misunderstood. Management believes that design is only the visual or interface design, but instead it is more than that. UX design involves customer research, usability research and when used correctly, it can be leveraged for requirements identification.

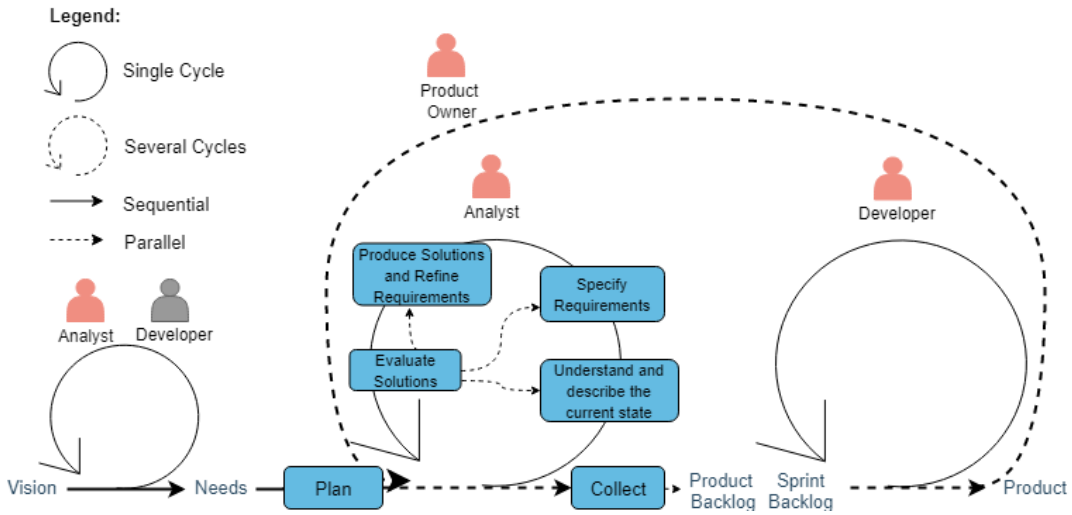

FIGURE 3.4: Human centered design process in Scrum adapted from Forbrig and Herczeg (2015, p. 7)

Forbrig and Herczeg (2015) provide the integration of Human Centered Design (HCD) with agile development using Scrum, as in Figure 3.4. The authors make use of three design patterns: **Sprint Zero**, used as a first cycle of analyzing the project

context and identifying needs. This sprint is performed by the Analyst, prior to actual software development (represented by the cycle from *Vision* to *Needs*, in Figure 3.4). The authors propose the idea that the process would benefit from the inclusion of developers (reason for modeled with a grey color in the picture); **One Sprint ahead**, used for requirements identification in the form of product backlog. This sprint is always performed one iteration before the development sprint (represented by the cycle from *Needs* to *Product Backlog*, in the Figure); **Parallel Tracks**, the development sprint and the requirements identification sprint are performed in parallel (with the requirements identification being one sprint ahead) and is iterated throughout the whole project duration (represented by the dashed cycle from *Needs* to *Product*).

3.5 Chapter summary

This mapping study contributed to acquiring a better understanding of the needs and motives for organizations to adopt Agile and DevOps practices. The benefits of Software Engineering artifacts traceability in agile methods has been identified and the potential interconnection of DevOps with process modeling has been explored. As a result, several mechanisms and practices for making organizations more robust and responsive to the dynamic market changes have been identified. Also, the alignment of the three units Business, Development and Operations was tackled.

The main conclusions from the Literature Review are highlighted below:

a) Organizations embrace DevOps and Agile methods, as they promote a cross-organizational collaboration and communication and contribute to faster delivery. Therefore, the combination of Agile and DevOps principles constitutes to a powerful solution for organizations.

b) A customer-oriented approach is prevalent, that DevOps has as a primary objective to satisfy those needs in a timely manner. Organizations that seek for being competitive and innovative have turned their focus towards leveraging customer feedback, by implementing it in the development of software increments, as fast as possible.

c) Bringing agility in the whole organization is a necessity for organizations to gain full advantage of agile principles. Organizations need support in scaling agile practices and achieving continuous process improvement.

The Systematic mapping study provided limited knowledge on the mechanisms used for communication and collaboration between different business stakeholders and the DevOps team. The role of PO is discussed in literature as the facilitator of BizDev and a high interaction of PO's role in the DevOps team brings better alignment between business and DevOps team goals.

The main concepts discussed in the literature review and their interrelation has been modeled in Figure 3.5 in the form of a Semantic net, following a similar approach and notation to Bock et al. (2014). As presented in the Semantic net, organizations have as a primary goal delivering value to their customers. Agile methods and the DevOps concepts have been adopted, in order to achieve this goal. As a result, organizations have to adapt their way of working in order to be more agile. The DevOps way of working constitutes of processes, that have to be executed in an efficient and optimized way in order to transform an idea or requirement into value delivered to the customer; people who perform those processes; and the technology

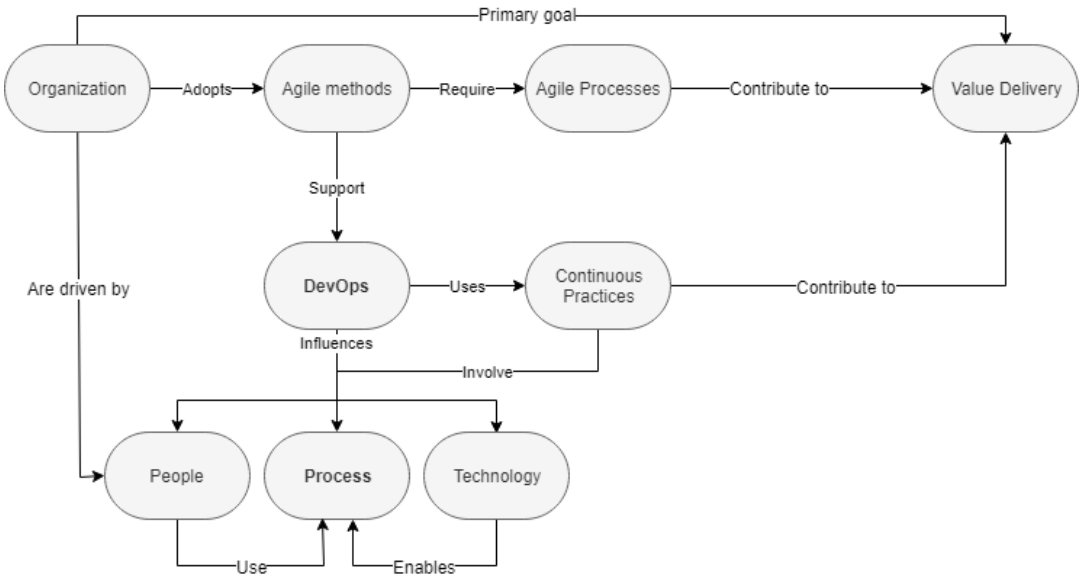

FIGURE 3.5: Conceptual Model of this thesis: Semantic net of Literature Review main concepts and their relationship

in place to support the processes. Bringing agility in the organizations processes and adopting continuous SE practices have been claimed to enable the achievement of organization's primary goal.

The Semantic net emphasizes the concepts DevOps and Process. The reason is to narrow down the focus of this research in the process level. The Semantic net provides a holistic representation of an organization leveraging Agile and DevOps as philosophies to achieve their operational goal. As the primary focus of this research is to contribute to a better Business and DevOps alignment, it is necessary to point out the different levels of alignment that can be achieved and how this research contributes to it.

Organizations use different methods to elicit and specify new requirements. This is a critical activity, if we consider that one of the reasons that development project fail is because of wring requirements (Forbrig, 2016b). Especially, in market-driven situations, requirements elicitation and prioritization becomes more difficult, due to the dynamic changes of the market and end-user needs. Traditionally, organizations use methods like up-front and long-term requirements planning. However, in the beginning of a project, many requirement are still unclear. This comes with the disadvantage of the business stakeholders usually making implicit assumptions about the new requirements and not frequently reviewing or changing them. The end-user's voice is not active in this process and this brings a major impact on the quality of the product delivered, which is not meeting the end-user needs (Olsson, 2018).

Although the benefits of adopting a systematic approach to maintaining the evolution trace of software engineering artifacts through traceability links are discussed, the current research will not continue further with this concept. The main reason is the different level of abstraction present in the traceability related concepts and the software delivery activities using DevOps. It is argued that this exploration was necessary as background knowledge for the current stage of this research. Moreover, this study identified the importance of *feedback-requirements* traceability as an important link to be obtained, for impact analysis of new features implementation. It is argued that maintaining this traceability link is very powerful, in the context

of this research. However, due to the following reasons, the decision ti discontinue with the concept was made: ***a)*** Traceability would cover a small part of providing a solution to the research problem; ***b)*** Traceability concepts are already implemented in the current DevOps process. Traceability tools are supported by the integrated DevOps toolchain.

Chapter 4

BizDevOps Industry Perspective

This chapter highlights the practical perspective of scaling Agile and DevOps practices across the whole organization and the role of the Business unit in facilitating this alignment. For this purpose, the results from the semi-structured interviews with practitioners are reported. The discussion in this chapter provides the first steps towards the artifact design, by combining both literature and industry perspectives on the research problem.

4.1 Company description

This research is conducted in collaboration with Accenture. Accenture is a global service company, providing its clients with a wide range of IT services and solutions in strategy, consulting, technology, operations and security.

Agile and DevOps adoption is of great interest to the company and new insights to support clients in this journey are highly needed. The interviews participants have been involved in DevOps related projects in different types of organizations and they were able to speak from own experience in different client situations and contexts.

4.2 DevOps in practice

DevOps is perceived as a trend in industry, that organizations would want to adopt in order to perform better and deliver faster. While in literature, there is no single definition for DevOps, also industry experts note this diversity in opinions. DevOps can be defined as a *mindset* and a *set of practices* combined with the power of technology.

"I think DevOps is one of those mindsets, specifically for helping organizations get to grips with the different silos they might have in the organizations, different technologies that they might need to employ in their organizations." (iv2).

Different perspectives emerge regarding the term BizDevOps. Prior to conducting the interviews, it was assumed, based on the literature search, that there is a thin line separating Business from software development. Therefore, the question on BizDevOps understanding was asked to foster practitioner's critical reasoning on the current DevOps and business alignment. Practitioners were skeptic about the added value of "Biz" in DevOps, as it is currently approached in practice. Table 4.1 summarizes all practitioners understanding and definition of BizDevOps. The main highlight is that the term itself does not differ much from what agile and DevOps advocate and that a first understanding is a closer collaboration between business stakeholders and the DevOps team.

TABLE 4.1: Practitioners definition of BizDevOps

Participant	BizDevOps understanding
iv1	Business strategy included into DevOps.
iv2	The same as DevOps.
iv3	I haven't heard of the term.
iv4	The collaboration between business, development and operations, including the quality (testing). The end goal is the same as DevOps.
iv5	I see the figure of UX the key part of BizDevOps. Business should be more customer than stakeholders.
iv6	The inclusion of Business into the team and how does that change the role of the PO, who is the voice of the customer.
iv7	The same as DevOps.
iv8	The same as DevOps. We have one team maintaining, owning and delivering new functionality on the product they are responsible for.
iv9	BizDevOps is shedding light to how you features, end users are doing and how the software is running, to give insights to different people who have shared responsibility in providing service to the end users. This is done though collecting data along the complete value stream of software creation.
iv10	BizDevOps is an intent to close the loop of feedback from the outside world (customers/market), by having the business in the team.

Another interview highlight is the practitioners perspective on the frameworks for scaling agile. SAFe is a popular framework used by many large organizations. In the literature review, it was found that frameworks for scaling agile practices support a good alignment of business unit with software development. Therefore, practitioner's perspective was asked, in order to identify how the SAFe framework contributes to a better alignment and what is the added value of adopting this framework.

Four of the interviewees were certified SAFe practitioners. They shared the view that currently, there is a high need to spread the cross-organizational importance of DevOps and how organizations can utilize its values and principles beyond software development activities (iv2, iv3, iv6, iv8). The concept that an organization is a value-stream[1], which needs to be optimized in order to meet the customer needs is emphasized there (iv2). Furthermore, an important point was that organizations tend to adopt a hybrid agile approach to different methods, in order to find the best fit for their needs. The need of combining and adopting several agile methods like combining Scrum, Lean and Kanban is mentioned by (iv1).

Although there are frameworks that guide an agile and DevOps way of working, organizations fail to benefit from their full potential. Moreover, from the interviews, it was observed that there is a big gap between what the frameworks or practices impose with what is happening in practice. Evidence for this observation provides the

[1] A value stream is described as long-lived series of system definition, development and deployment process steps, which is utilized to develop and deploy systems that supply a constant flow of value to the business, customer or end user (Alqudah and Razali, 2016, p. 831).

fact that all the interviewees were constantly referring to "an ideal situation" versus "what is happening in reality". In our opinion, the main reason is the organization's lack of DevOps and agile maturity. However, as this is not a primary concern in our research, we intend to solely observe the current situation in practice and identify the most important challenges.

4.3 Alignment challenges

The main contribution of the first round of interviews is a better understanding of DevOps in practice and the identification of challenges that agile organizations face when adopting DevOps.

Organizations, regardless of their size, are facing challenges with software delivery, such as not delivering smoothly enough, not delivering value to the customer in a timely fashion, mentioned by (iv2). *"One of the biggest challenges today is how to scale DevOps across the whole organization, to achieve end-to-end DevOps and how to standardize this over a line of business or even at the enterprise level."*, mentions (iv4). These challenges have different nature. In order to acquire a structured representation, three categories have been used to group these challenges, namely *Business challenges*, *Organizational challenges*

This research adopts the view of the organization that seeks for innovation. In CSE, Continuous Innovation is tightly associated with Continuous Experimentation. Fabijan et al. (2017) provide an Experimentation Evolution Model for guiding practitioners to scale Continuous Experimentation in software organizations. In the Experimentation Evolution Model, the authors present three dimensions of organization's evolution towards experimentation, the *technical*, *organizational* and *business* dimension. Olsson (2018) adopt the three dimensions and define several challenges associated with them. In this research, it is relevant to use the same categorization for discussing the challenges associated with the DevOps and business alignment. The two categories are defined below (Olsson, 2018):

- **Business challenges** refer to challenges faced in the business side, related to goals alignment with the development team, evaluation metrics that are in place and how decision making is enabled.
- **Organizational challenges** refer to challenges faced by the cultural aspect of the organization, the organization hierarchy and the different roles involved in the delivery lifecycle.

4.3.1 Business challenges

Understanding business value

One of the major challenges in the agile world is requirements prioritization, due to the inability to properly understand and measure business value. The role of the PO is the one bridging the alignment gap and should be the one to understand the immediate business value of a requirement. There is a constant need for prioritization and balance between pushing new requirements to development and dealing with incidents and Service Level Agreement (SLA) monitoring (iv3).

Lack of business metrics

In a typical DevOps situation, there are several performance metrics that the team is interested in reviewing and taking into account for the next iteration. Metrics that are usually in place are related to the team health and customer-centred metrics (iv2). The DevOps team is less interested in business KPIs (iv3) and business performance related KPIs are missing in the delivery lifecycle (iv4).

Long term Planning

The Business planning activity has a frequency of 3 months and this brings the risk of having outdated list of requirements in the product backlog. This traditional way of the business slaying a long term plan and never coming back to it for review and evaluation on the way is one of the main challenges of working agile. In today's market, customer requirements are changing dynamically and competitive organizations must find a way to continuously implement those changes. Companies like to work in the old-fashioned way like large planning, where the DevOps team is usually not involved. It is then a project management thing, to be transferred to the DevOps team and they tend to follow the product backlog, even though some user stories are complex (iv5).

4.3.2 Organizational challenges

Culture and mindset

Culture and mindset play a very important role in the successful adoption of agile and DevOps, yet it is a big challenge. This mindset is very hard to change in the higher managerial level, where people are used to the traditional way of working, expecting deliverables in a long-planned basis (iv2, iv3, iv4). Usually, the mindset is characterized by a "us versus them" mentality (iv3).

Delivery Blockers

Business activities, development and operations processes have different occurrence frequency. In a typical Scrum team, that works in a DevOps way, process automation makes the delivery pipeline run smoothly, but there are often tests (like user acceptance test) (iv4) that need to be performed and approved by business stakeholders in order to proceed in the next (production) environment, the production environment. This is causing a lag in fast delivery (iv1).

Timely reporting

While agile methodology supports that there is no need for systematic documentation and measuring, the management level requests for reports and performance results based on numbers and metrics are contradicting this agile principle (iv3). One way to achieve business alignment with the software delivery lifecycle is via shared dashboards and making everyone aware of metrics and performance KPIs (iv4). Updating business about the entire lifeycle, updating on the status of the release and the requirements, in order to achieve alignment on what is expected and what is actually delivered.

Role of the Product Owner

The role of the PO has been identified as an important enabler for a successful alignment of the business and development perspectives. However, practitioner mention that there are usually several challenges associated with the quality of this role. Often, in practice the figure of the PO is disconnected from the team and is a not dedicated role. This is causing the team to loose the business representative (iv1). Also practitioners mention the influence of internal politics in the role of the PO (iv2). The quality of the PO is sometimes influenced by the most powerful stakeholder or the 'one with the loudest voice'.

4.4 DevOps Stakeholders analysis

An important part of this research is understanding the different stakeholders involved in the software delivery lifecycle and how is the distribution of roles and responsibilities between them. As identified earlier, DevOps adoption differs according to the organization needs, size and product delivered. Therefore, the roles involved in the delivery cycle also vary on the project needs. There are multiple stakeholders involved in the delivery process. The following stakeholders categories are identified:

1. Management: Representing the business voice of the agile organization, responsible for providing domain expertise, steering and keeping track of the software project progress.

2. Product Owner: The PO is responsible for communicating the business requirements to the DevOps team. He/she has a very good understanding of the product and he has the ability to transfer the business perspective into the software development perspective.

3. DevOps team: The DevOps team consist of developers, testers, operations roles, infrastructure architects. The Scrum master is also part of the team.

4. Customer: An observation on a twofold understanding of the role of the Customer was made from the expert interviews. Two types of *Customers* are identified in organizations adopting DevOps:

   a) Internal customer: The Management (or the business) is the customer of the DevOps team effort. The Management is setting the project plan, consisting of milestones and deadlines for the major releases. This plan, which is usually a long-term, fixed plan is translated to the development team through the PO. The DevOps team is working on small agile iterations in order to satisfy the predefined plan.

   b) External customer: The end-user is the customer of the Business and DevOps team combined effort. This is the situation, where the end-user is the primary focus of both the business and DevOps team effort.

Practitioners define agile maturity based on more subjective capabilities, such as collaboration, communication, commitment, care, sharing, self-organization rather thane in numbers, metrics and processes (Fontana et al., 2014). The authors present the relationships of the concepts that define maturity in agile software development, as in Figure 4.1

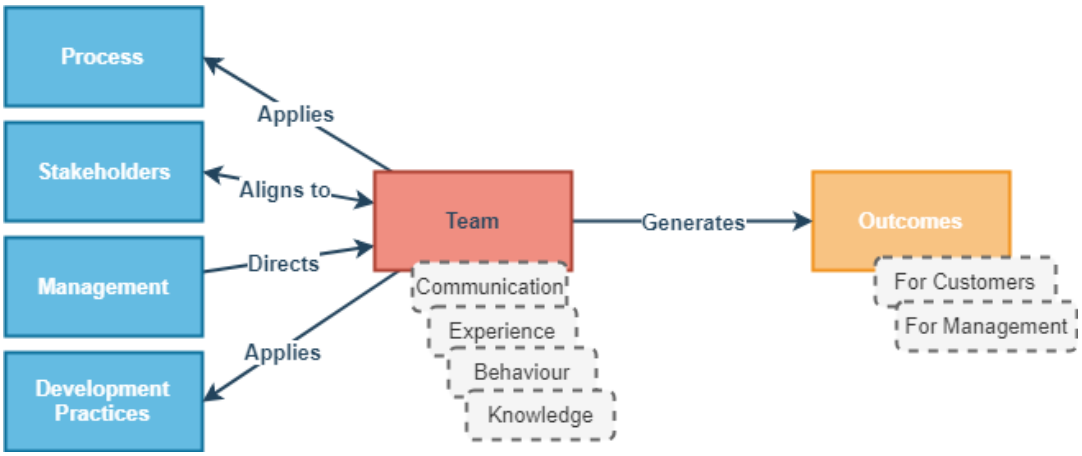

FIGURE 4.1: Concepts that define agile software development maturity and their relationships, adapted from Fontana et al. (2014, p. 152)

In Figure 4.1, the twofold existence of a Customer, as discussed above is present. The Team generates Outcomes for the Customers (end-users) and for the Management. This situation is often causing added pressure and hierarchies to a smooth delivery, with the DevOps team being asked to satisfy two types of stakeholders. While the focus should constantly be on the end-user of the software product, the multiple stakeholder dependencies are causing bottlenecks and unmet user needs.

4.5 Literature and Industry combined perspective

The related literature discussion provides a starting point for the research gap exploration and presents evidence on missing scientific contributions on the Business and DevOps goals alignment problem. The main contribution of interviews was to provide the contextual insights of business involvement in the software development lifecycle.

As a result, several *business* and *organizational* challenges have been identified, that prove the need for better collaboration and communication between the business perspective and software development in agile organizations adopting DevOps. A successful process of capturing and prioritizing the right requirements is thought to have a big impact on aligning the business world and the software delivery cycle (Lehtola et al., 2009).

Organizations adopt DevOps to achieve faster delivery time and higher value delivery to the customer. Currently, organizations adopting DevOps are still facing several challenges:

1. There is not a common understanding of what is DevOps. Organizations must understand that it is more than just a team reorganization with Development and Operations sharing responsibilities, but that the whole organization should work with the same mindset.

2. Organizations should be driven by the concept of highest value and prioritize requirements and tasks based on value criteria. However, since there is not a single way to define business value, it is hard to determine whether the same understanding is perceived both in the DevOps team and the business side. This is the root cause of the goals misalignment problem.

3. Customer-centricity is one of the core Agile values. However, we identify that there are currently challenges in maintaining this bond. Furthermore, current studies on BizDevOps put an emphasis on a Human Centered Design integration with agile development methods (Forbrig and Herczeg, 2015; Forbrig and Dittmar, 2019; Forbrig, 2018b; Forbrig, 2018a). Eliciting customer requirements is one major challenge in the current software delivery processes. Practitioners mention that in the DevOps team level, there is usually very little interaction with the end-user, or it is completely missing.

4.6 Conceptual framework in a nutshell

In order to provide the reader with a consistent view of the research, it is necessary to define several concepts that are used. For this reason, the remainder of this report builds upon the following definitions:

Business: In this research, a Requirements Engineering point of view is adopted. Business (often the Biz abbreviation is used) refers to the people and activities involved with providing input for the Software Development Lifecycle. The input takes the form of requirements, regardless of their specification method. Also, Business is responsible for capturing the customer needs and translating them into new requirements.

Business Goals: The software product development is a means of achieving a set of pre-defined business goals. An organization uses several mechanisms for defining goals. Translating the organizations vision into goals is one of them. The goals can have different scope and they are highly dependent on external market factors and internal factors like continuous improvement initiatives. In this research, we define business goals as the primary source of requirements for the software development process. Therefore, a logical rationale would be that the business goals are translated into requirements in the product backlog. No further elaboration is necessary on defining the nature of these goals, as it is out of scope for this research and we aim for generalizing the approach.

Research gap: The gap addressed in this research refers to the challenges that Business stakeholders face in collaborating with the DevOps team. The gap takes the form of misalignment in goals and expectations between the two groups of stakeholders.

BizDevOps: Is the problem of alignment between Business, Development and Operations, with the intention to close the delivery cycle, or discontinuity present by several *business* and *organizational* challenges. Therefore, the BizDevOps process designed in this research aims to propose a way of working in agile teams and supports the role of the PO in the task of requirements elicitation, prioritization and monitoring until their fulfillment. The BizDevOps cycle ends with the validation of the requirements fulfillment.

Definition of Continuity: With DevOps, there is a big emphasis on speed, by automating a big part of the delivery process. However, Continuous Software Engineering advocates continuity and flow as more important than merely achieving high delivery speed (Fitzgerald and Stol, 2017). A system that emphasizes continuity and flow can benefit in several ways: ***a)*** flexibility and rapid adaptation ***b)*** improved quality and resilience of the software

Continuous Business Alignment Definition: Since we adapt a process-oriented view of DevOps, the goal of Biz and DevOps alignment implies a form of *organizational process alignment*. Literature defines *organizational process alignment* as a capability to arrange various parts of a company to work together harmoniously, in order to pursue common organizational goals, enhance performance and sustain competitive advantage (Hung et al., 2007). Some important aspects to linking all areas of an organization are: having a cross-functional organizational structure (Weiser, 2000) and continuously focusing on customers and their changing requirements (Hall, 2002). Hung et al. (2007) state that *organizational process alignment* can be interpreted as the effort required to make processes the platform for organizational structure, strategic planning and information technology. Furthermore, a successful organizational alignment positively influences business performance (Hung et al., 2007).

The aspect of continuity means that the software development projects do not come to an end, when work is completed. On the contrary, the focus on what needs to be done is changing (Forbrig, 2016a). For example, in the beginning of a project more emphasis is given to requirements gathering and design, later on development and later on software maintenance.

In order to gain DevOps full potential, an organization should focus first on the needs of the business and then aligning DevOps (Ravichandran et al., 2016). Doing DevOps just for the sake of it, can lead to misalignment of goals and if it is not serving the needs of the business, than the gap becomes even bigger.

Therefore, we aim to reinforce the importance of *organizational process alignment*, which in this research is referred to as *continuous business alignment* in the way organizations apply the DevOps way of working.

Figure 4.2 presents the scientific foundation of how the desired Biz and DevOps alignment can be achieved.

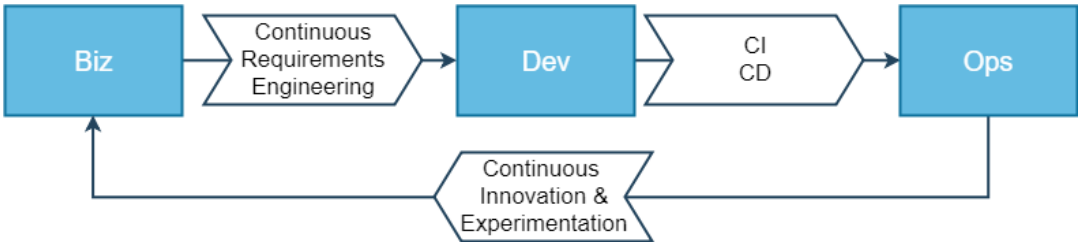

FIGURE 4.2: The big picture of this research: Bridging Business and DevOps with the use of Continuous Software Engineering practices from Fitzgerald and Stol (2017)

4.7 Chapter summary

This chapter provides the overall picture on the research problem, acquired through the *Problem Investigation* phase. The combined results from literature search and expert interviews provide the background for the next phase of *Treatment Design*. Moreover, for the purpose of clarity, the conceptual framework of the research is presented, where necessary concepts are defined.

Chapter 5

Artifact Design

"An artifact is something created by people for some practical purpose." Wieringa (2014, p. 29). Artifacts in information systems and software engineering are studied and designed in the context of their use. This chapter explains the design of the BizDevOps process model, as the primary deliverable of this study. The Chapter starts with discussing the requirements for the treatment solution (Section 5.1), later dives into the rationale behind the design choices (Section 5.2) and ends with a description of the BizDevOps process model components (Section 5.3).

5.1 Artifact Requirements

A list of requirements for the treatment solution has been derived, based on the results from the *Problem Investigation* phase, as described in Chapters 3 and 4. The process of analyzing all the insights gathered from Literature search and interviews and drawing conclusions for proposing the BizDevOps process model, as a solution treatment is based on the rationale described in Section 2.3.

The following requirements, presented in Table 5.1 serve as a starting point for the BizDevOps process model. The requirements are supported by literature and interviews findings and are further explained later in text.

TABLE 5.1: Requirements for the BizDevOps process model

Nr.	Requirement	Literature	Interviews
R1	A high-level process description is needed to guide the implementation of a user-centered process in BizDevOps.	(Ferre et al., 2005), (Forbrig and Herczeg, 2015)	
R2	The process should be supported by frequent feedback loops.	(Dobrigkeit et al., 2019), (Häger et al., 2015)	(iv10) (iv5)
R3	A definition of the agile team roles and responsibilities that take part in the process should be established.	(Schrader and Droegehorn, 2018), (Wiedemann et al., 2019)	(iv5)(iv6)(iv10)
R4	Performance checkpoints should be in place, to continuously measure performance.	(Dobrigkeit et al., 2019), (Häger et al., 2015), (Heidenberg et al., 2012)	(iv10) (iv8) (iv6) (iv5)

R1: A high-level process description is needed to guide the implementation of a user-centered process in BizDevOps.

We argue that by specifying a process that has a human-centered approach to software development, organizations can improve their requirements elicitation process and become more user oriented. This research adopts the view of the organization as a System of Innovation, where organizations are interested in establishing a sustainable and responsive attitude towards the evolving market conditions and the end-user needs (Freedman, 2016). For establishing such a system, a series of Continuous Software Engineering practices can be utilized from Fitzgerald and Stol (2017), Fitzgerald and Stol (2014), and Theunissen and Van Heesch (2017). The Continuous Planning activity facilitates the BizDev connection, as it enables tighter connection between planning and execution (Fitzgerald and Stol, 2014). In our process model, this tighter connection is facilitated by a successful flow of requirements from Biz to Dev and by jointly contributing to requirements elicitation.

As a result, software development projects can start with a better understanding of user requirements and this can reduce the risk of building software products that do not meet the user needs.

This requirement is only supported by literature, because there was no direct finding from the interviews that highlights the integration of user-centered process in BizDevOps. Although, iv7 and iv5 emphasize the importance of end-user involvement in agile development and DevOps process, this inference do not completely match with R1. Therefore, the R1 originates solely from related works on BizDevOps.

R2: The process should be supported by frequent feedback loops.

The BizDevOps process model aims to enable a closer alignment of DevOps team with Business stakeholders. Therefore, a continuous feedback mechanism can facilitate the alignment of goals between different stakeholder groups and enable continuous learning in the organization. Continuous feedback is necessary in two directions, between the DevOps team and business stakeholders and early feedback from the end-user.

"The problem is that most of the companies work Agile, but they lack feedback loops that are necessary to detect changes in the outside world for example customer interviews, market research. If you detect changes they should be fed back to the development system and this feedback connects back to the customer."(iv10).

R3: A definition of the agile team roles and responsibilities that take part in the process should be established.

One of the main challenges in agile software development relates to the alignment of stakeholders and users in the development process. In particular, it is hard to engage all stakeholders to participate in the whole development process, to understand that the development team can make independent decisions (Schön et al., 2017) and there is a lack of understanding from the business side on how to properly address the interaction with the customer (Schrader and Droegehorn, 2018). Therefore, for the BizDevOps process model, a reorganization of the agile team is necessary and should be explicitly defined.

R4: Performance checkpoints should be in place, to continuously measure performance.

In practice, it is often seen that organizations do not validate their business case. While in the beginning of the project, there are some target outcomes, at the end, no validation on the outcomes or business benefits is measured and evaluated. This brings a big disconnect in the business alignment with software delivery cycle, which hinders the ability of the business to steer the agile organization based one measurable performance outcomes.

"Understanding business value is difficult topic for teams, or how to measure it. Each company has different business value that needs to be realized and how are they are going to measure it, otherwise how can you make sure that you are growing. You get that understanding when you have a more mature team."(iv8)

Measuring the realized business value can help make better decisions (Heidenberg et al., 2012). Therefore, end to end performance checkpoints should be in place, for the BizDevOps process model.

5.2 Towards designing the artifact

The alignment problem addressed in this research consists of the long-existent challenges in integration of Business and Development domains, that are referred to as *scopes*, in this research. Additionally, two *units* of alignment are considered, the people and process alignment. Figure 5.1 presents the desired integration and what means are used to facilitate it.

Business scope: involves the activities, artifacts, actors and the technology involved in business planning of the software product. This research is focusing only on the requirements engineering process (budgeting and other related business planning activities is out of scope). In this scope, the business goals are set by the business stakeholders.

Development scope: involves the activities, artifacts, actors and technology involved in product delivery using DevOps. In this research, the Scrum method is adopted, therefore the Development scope can be envisioned ad the Sprint cycle. The development team has its own goals, which are the translation from the business goals.

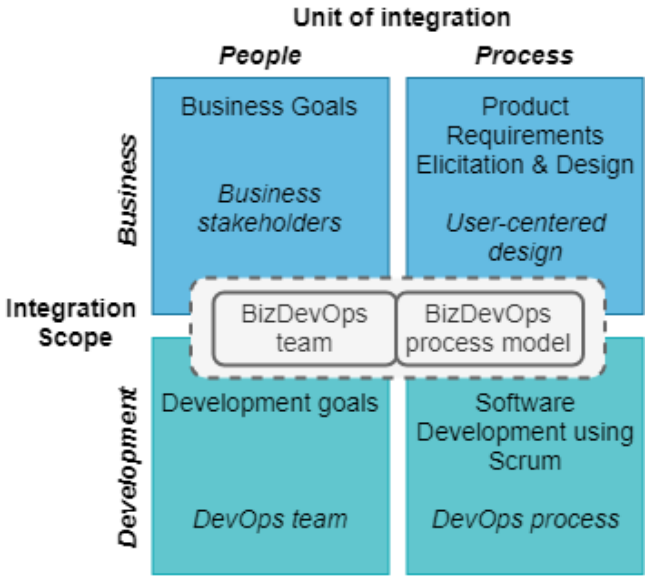

FIGURE 5.1: Visualization of the two Integration *Scopes* and *Units* of integration for the achievement of the desired alignment

Integrating the business and development scopes: For integrating the two *Scopes*, the two *Units* of integration are considered, namely *People and Process*. The alignment of *People*, would contribute to both business and DevOps teams working on the same

goals. This research achieves this alignment, by introducing a form of team organization, the BizDevOps team. For the alignment of processes, the different agile methods discussed in Literature Review (Chapter 3) have been leveraged.

5.2.1 Artifact design process

The BizDevOps process model is a construct that inherits concepts from different agile methods. In order to be able to assemble the concepts and procedures from the different methods components, the method engineering approach from Goldkuhl et al. (1998) is used. Goldkuhl et al. (1998) describes that for integrating a method, one can make use of several method components and the decision on which method components to use is dependent on the situation. Figure 5.2 presents the main notions for the realization of a method integration.

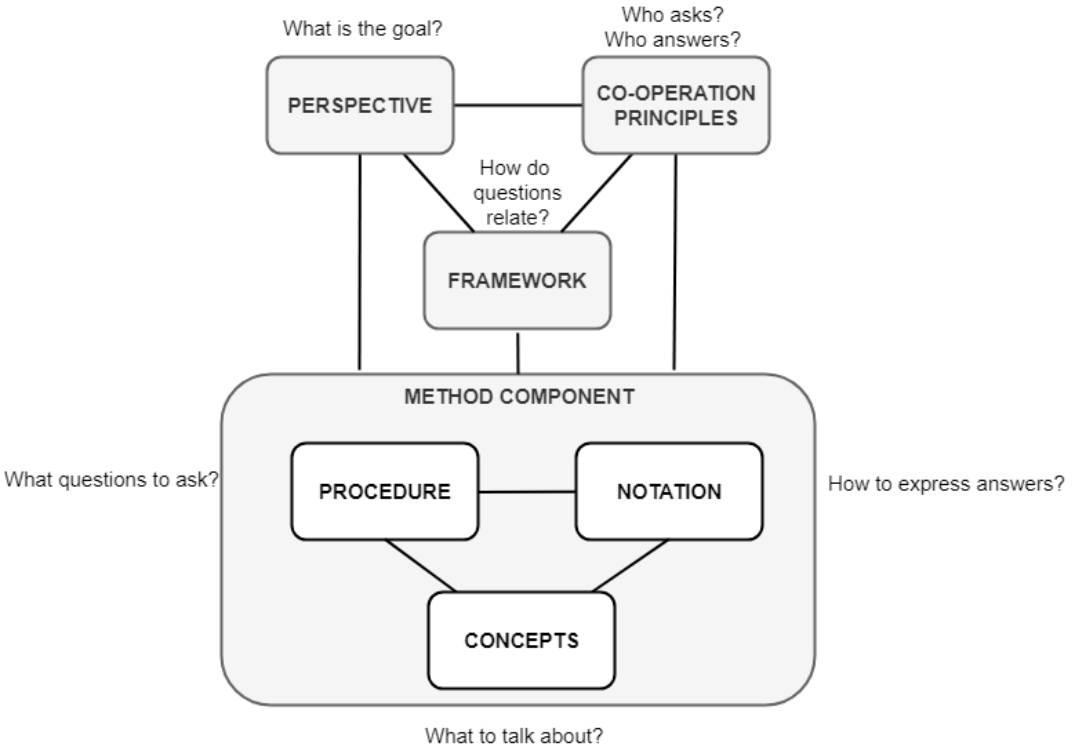

FIGURE 5.2: Method conceptualization framework, adapted from Goldkuhl et al. (1998)

Perspective: represents the goal, purpose (explained in Section 5.2) and the preconditions (explained in Section 5.3.1) for applying the BizDevOps process model.

Framework: gives an overview of the main concepts used for the process model (represented by the phases, activities, transition points and input in the Figure 5.3).

Co-operation Principles: consist of the description of the roles and actors that are necessary to perform the process and the functions in the BizDevOps team (explained in Section 5.4.1).

Method component: consists of three elements: ***a)*** **Procedure** describes the main process phases, concepts involved, describes the high-level process flow and the transition between the process phases (explained in Sections 5.4 to 5.6); ***b)*** **Notation** specifies the way of documenting the results from the BizDevOps process model activities; ***c)*** **Concepts** specify the aspects that are necessary to integrate the procedure and the notation, in order to construct the BizDevOps process model.

The method conceptualization framework from Goldkuhl et al. (1998) provides a basis for conceptualizing the nature of the designed treatment and enabled the assembly of different agile method components. However, it is important to note the distinction and the choice for designing a process model.

"A process model is a representation of a process, describing how the process is performed (descriptive model) or how it is expected to be performed (prescriptive model) at a selected level of granularity." Kneuper (2018, p. 2).

A model has three properties, according to (Kneuper, 2018):

1. Mapping property: a model is a mapping or a representation of some entity (the software delivery process, in this research)
2. Reduction property: a model does not contain all the attributes of the original entity (in this research, a high abstraction level is adopted)
3. Pragmatic property: a model is created for a certain purpose (in this research, the software delivery process is constrained by the research goal and scope)

Due to the above mentioned properties, a process model provides an explicit understanding of the process, while at the same time allows for enough flexibility, when applied by the agile organization.

5.3 The BizDevOps process model

The BizDevOps process model, presented in Figure 5.3, consists of three main phases, that represent the high level, core activities for the software delivery process. The arrows represent the transitions from one phase to the other. Each transition indicates what goes as input from one phase to the next one. Due to the fact that in agile software development, there is not a strict sequence of steps to be followed, but rather the processes are highly iterative, the BizDevOps process follows the same principles. The purpose is to not restrict the workflow with a pre-described sequence of activities to be performed, but allow flexibility in the process model steps, that can be adjusted to different agile project needs. For example, in a highly innovative project context, the team might want to iterate several times on the *Explore and Identify* phase, before jumping to actual software development. Each phase is later expanded into a series of iterative activities (in the bottom of the model), where the most important concepts and artifacts for each activity are highlighted.

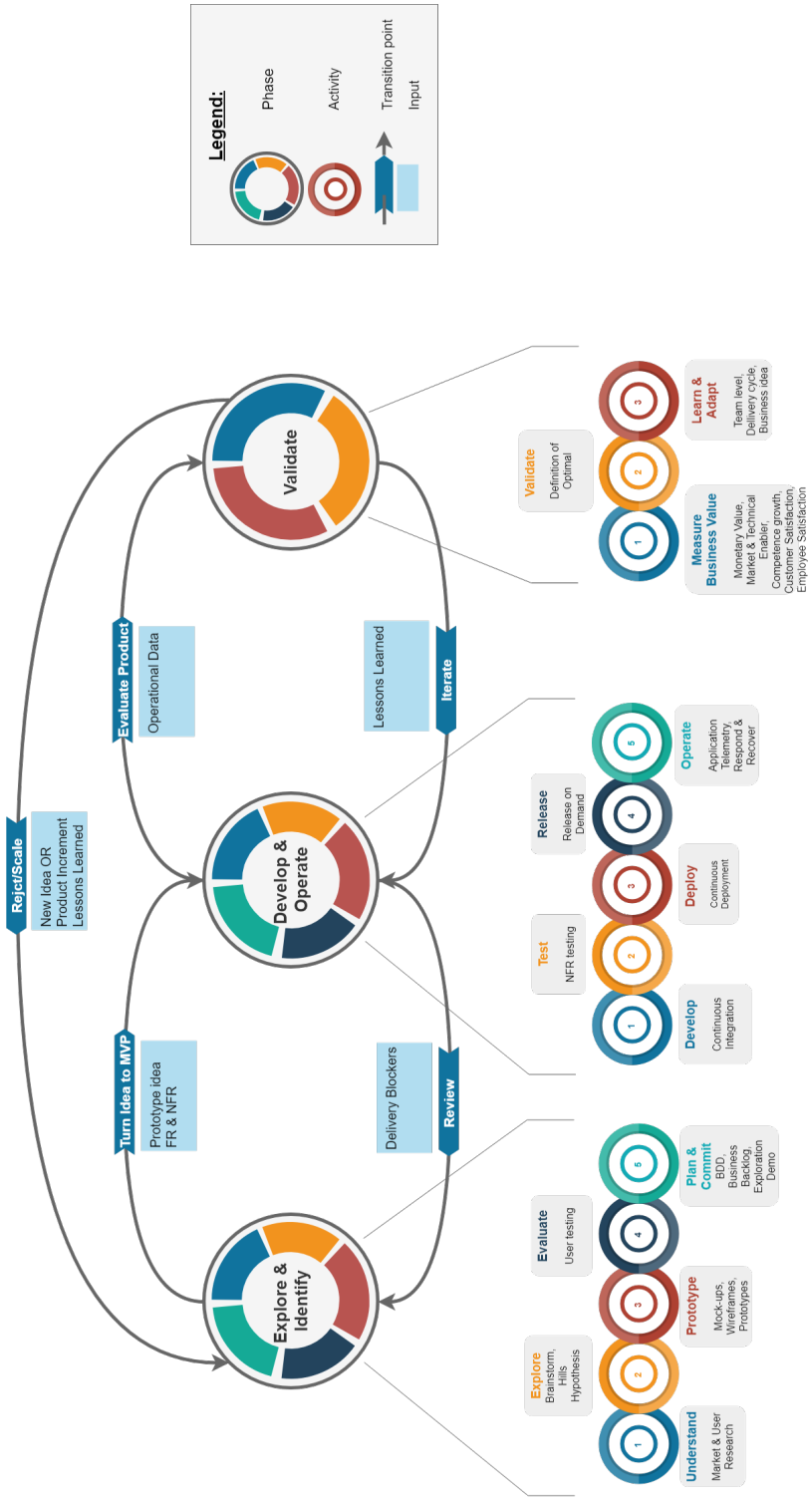

FIGURE 5.3: The BizDevOps process model

The following sections elaborate on each of the three phases and provide a thorough description of the activities for applying the BizDevOps process model. Each section has the following structure:

a) first, a description of the goal and purpose of the phase is provided

b) next, the activities that take place in each phase, together with the core artifacts are explained

5.3.1 Prerequisites for the application of the process model

Due to the fact that the BizDevOps process model intends to extend upon existing DevOps models or frameworks, some preconditions for the agile organization that applies the designed BizDevOps process model are necessary.

First, the agile organization should have a certain level of DevOps maturity in place. Due to the fact that in our scope, the organizations transition to Agile and DevOps is assumed as a prerequisite. Therefore, later in the description of the artifact, the software development process using DevOps is not elaborated in detail.

For better comprehension of the desired maturity level, a short description is provided (as in Table 5.2), that relates to the DevOps maturity matrix from Feijter et al. (2018). The maturity evaluation has been performed based on the researcher's knowledge acquired from the interviews and literature. Table 5.2 is structured according to three main perspectives of DevOps maturity, *Culture and collaboration*, *Product, Process and Quality* and *Foundation*. As a result, an organization that makes use of the BizDevOps process model has a desired DevOps maturity of level 8, 9 or 10 (in a scale from 1 to 10).

TABLE 5.2: Organization's desired DevOps maturity description

Perspective	**Description**	**Maturity**
Culture and Collaboration	Structured interdisciplinary communication is in place within the DevOps team and between the team and management. Active knowledge sharing takes place between professionals. A culture of trust and respect is a core value. The DevOps teams are cross-functional and T-shaped.	9
Product, Process and Quality	Experiments with features are run systematically, for supporting backlog prioritization. Continuous Integration of software builds, automated code quality monitoring and automated systematic testing is in place. Continuous Deployment and automated rollbacks are possible. Automated material generation for Release to Production. The identification of root cause of incidents is supported by analytics	10
Foundation	Version Control Configuration Management. Software and architecture evolves continuously and is aligned. Platform as a Service infrastructure.	8

Second precondition, the development team in the agile organization is working with the Scrum method. This precondition is however, not restrictive towards other agile methods. It is rather used for easier understanding and reference to the different ceremonies, artifacts and roles in the software development process. Scrum

provides a structured process to agile development and a commonly understood jargon in the academia and industry, by making use of the the Scrum ceremonies, roles and definitions. Therefore, it is relevant to adopt a Scrum-based DevOps process in our further explanation.

5.4 Phase one: Explore and Identify

The Explore and Identify phase is a human-centered process to requirements elicitation and product design for the software product to be developed or under development. This phase is closely related to the *business scope* and how requirements are identified and prepared for the upcoming phase of agile development using DevOps. The Explore and Identify phase provides a bridge between product design and product development. This phase uses concepts from user-centered process and Design Thinking, and methods for innovation like Lean Startup and Lean UX, for the purpose of product discovery.

The purpose of this phase is to identify product ideas that bring value to the business and specify requirements in an innovative way. Linking back to the research problem, this phase is where the business goals are translated to more tangible artifacts, in the form of product requirements and where the first point of business and DevOps alignment goals take place. The main rationale is to enable a continuous process, that ensures that the software development effort (the next phase of the process model) is entirely focused on the needs of the business and is not merely serving technical solutions and architectural rollbacks (Holtsnider et al., 2010).

To achieve the desired alignment, in the BizDevOps process model, organizations put a big emphasis on the project initiation and pre-development activities. Starting with the wrong requirements can lead to products that fail to meet the user needs. Therefore, identifying the right product requirements is very crucial for such a system. Investing time and resources for conceptualizing the problem can save costs of later unexpected failure (Holtsnider et al., 2010).

"It is important that we start with the right project, that the business case makes sense, that the design idea we have makes sense. It prevents you from making expensive decisions." (iv8).

The BizDevOps process model is suitable for organizations that are open to innovation and adopt a culture of "experimentation without regret". Organizations can make use of two types of innovation, namely Incremental ans Disruptive innovation. Incremental innovation or emergent innovation is driven by inside the organization. This type of innovation concerns improving performance of existing products along the dimensions that mainstream customer's value (Pozzey et al., 2012). This type of innovation happens when ideas about a feature product originate from the agile teams or any other related stakeholder. Customer feedback on existing products is a kind of internal innovation. Disruptive innovation radically transforms the organization and enables competitive advantage through delivering value to new markets (Pozzey et al., 2012). It originates from outside the organization, for example market needs, competitors, new technologies that can be exploited. In order to use disruptive innovation, the organization should make use of design-led tools. One example is when organizations are using their Innovation Hubs to get new ideas about their products.

While Incremental innovation is easier and a safer choice to be exploited, organizations are usually reluctant with applying Disruptive innovation, due to the high level of risk it is associated with. Empowering both types of Innovation is

crucial for organizations that want to stay competitive and deliver high value products (iv10). Therefore, the first phase of the BizDevOps process model describes the generic steps that describe how organizations can embrace product innovation, that occurs continuously and is incorporated as part of the organizational culture and daily operations.

It is important to distinguish two cases in the software product development lifecycle:

1. New product: When a new product is being developed, there is a higher need for an extensive application of the *Explore and Identify* phase. The product requirements are in the early stages, many requirements are still unknown and the need for understanding and eliciting user requirements through rapid prototyping and user feedback is bigger.

2. Existing product: In case of an existing product, there is no need for spending much time on the *Explore and Identify* phase. Instead there is a need to find the balance between eliciting new product ideas or features and maintaining the existing product.

Figure 5.4 shows the proportion between product exploration activities and software development, during the course of the Product lifecycle. The figure indicates that when the product is new, the team should invest more resources on the *Explore and Identify* phase, in order to create a better overview of the product requirements. As the product becomes more mature, the use of exploration activities is reduced. However, the process is continuous and iterative and the end life of the product indicates that the business might have decided to start a new product line, which brings the Product lifecycle back to its initial stages, with a high need for product ideas exploration and less development.

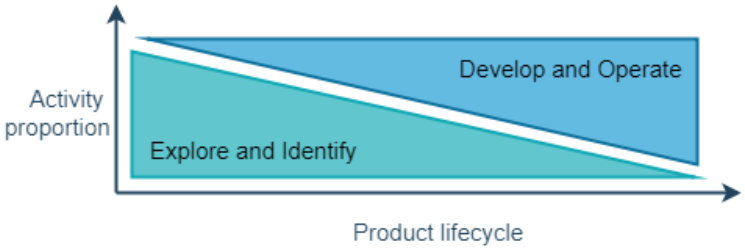

FIGURE 5.4: The proportion of applying the *Explore and Identify* phase and *Develop and Operate*, during the course of the Product lifecycle

One of the essential parts of the BizDevOps process is the human factor and the roles and functions that perform the process activities. This model proposes a reorganization of what is known as the DevOps team, a description of the BizDevOps team is part of the artifact and is essential for the application of the process.

5.4.1 The BizDevOps team

The BizDevOps team suggests a form of agile teams organization, where people from the three disciplines, business, development and operations work collaboratively during the entire software delivery lifecycle. This structure resembles a Feature team, defined as is long-lived, cross-functional, cross-component team that completes many end-to-end customer features (Scrum, 2018). This way the team is focused on optimizing delivery and maximizing customer value. The team has the

right competencies to be responsible from end to end for the feature. Because the BizDevOps team is multidisciplinary, consisting of representatives of the three units, business, development and operations, it is ensured that all the necessary skills are present in the team.

"Within this approach, the business team sets requirements, but they also work directly with developers "to set priorities for agile software development sprints and backlogs. They become partners with the business-side and work with managers to solve problems and achieve business goals." Wiedemann et al. (2019).

A visualization of the BizDevOps team functions is presented in Figure 5.5. Usually, the *Business* function is fulfilled by the PO. An important skill of the *Business* function is having technical knowledge, in order to understand and be able to mediate the communication between the business stakeholders, users and the development team. However, except from the PO, this research suggests that another representative from the business should be in regular contact with the team, depending on the needs of the project.

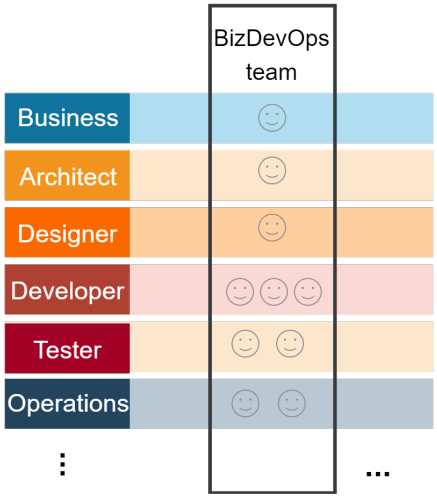

FIGURE 5.5: The BizDevOps team

"Adding business development people inside the team, that will work closely with the DevOps team and will always be available for feedback. They [business] explain to the team how the customers perceive the product, what they missed or need into the documentation. They make small specifications, they specify test cases. It worked very well, we call these Feature teams. They [business] could also put pressure on the company better than the development team could." (iv10).

Other roles that are necessary include: the *Architect* function, needed to provide the expertise on the technical implementation of the idea. Technical architects, Solution architects can fulfill this role; the *Designer* function (User Experience, User Interface) bring the software usability point of view in the team; *Developers* and *Testers*, that are part of the traditional Scrum team; and *Operations* functions, that enable software operation and maintenance.

BizDevOps team attributes:

One attribute of the BizDevOps team is *Security*. Team *Security* is associated with the readiness of each team member to be involved in activities that are not directly associated with his role and are not his specialization, for the sake of keeping the team

integrity. By collaborating and sharing knowledge, the team can build higher security that problems can be identified in early stages, and the output of the software delivery is meeting the business stakeholders and the customer needs. Therefore, the risk of rollback and potential rework is minimized.

Autonomy: In BizDevOps, the team is empowered so much that it can take independent decisions. The independency in decision making enables an even faster delivery process. The figure of the Business representative gives more power to this. The team has the Business point of contact that is available and ready to support, in case of important decisions, that require business approval. Due to being multidisciplinary, the BizDevOps team has the right balance of specialization and flexibility in the team functions.

Scalability: In large-scale projects, with multiple BizDevOps teams, the teams should be able to align with each other and not become blockers of each other. The fact that the teams are self-organized doesn't impose isolation form other teams. The BizDevOps teams are able to understand the needs of the work in progress and they are easily scalable, according to the workload demands.

The following subsections describe the activities performed in the *Explore and Identify* phase, presented in Figure 5.6, for the readers convenience. The BizDevOps team is performing the three phases of the process, but as mentioned above, with different levels of involvement. In the Explore and Identify phase, the Business role is more prevalent.

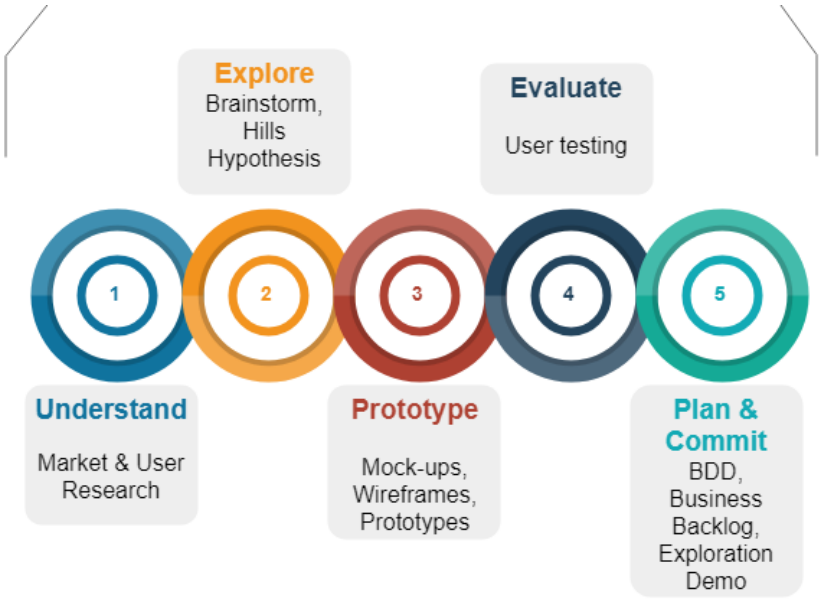

FIGURE 5.6: Phase1: Explore and Identify activities

5.4.2 Understand

The *Understand* activity initiates the requirements discovery process. The goal is to create a thorough understanding of the user of the software product and his/her context, by leveraging information from different sources, like market and user research. In this activity a customer problem or a market opportunity that has the potential for leading to innovation is being exploited.

Although the end customer is in focus of agile development, often the origin of requirements is the business stakeholders or the PO and there is not an implicit way to involving the customer voice in the requirements elicitation process.

"I like to work in a way that I would always use the end user. But this is not always feasible. I have never seen somebody building a software product and ask for the end users... We try to involve as many stakeholders as possible, so we try to think as an end user. When I used to work in Telecommunication companies, when the business needed to catch up with the market, to be more competitive, to create something, they would propose it, as seen from the others."(iv7).

5.4.3 Explore

After the BizDevOps team has created an understanding of the customer and market needs, the next step is exploring potential solution ideas, by brainstorming.

The results of the previous *Understand* activity are synthesized, in order to build a point of view (PoV) for the problem. In Lean startup method, the goal is to find the problem-solution fit, means to brainstorm different ideas that solve the customer problem, with the intention to test and validate, it in the later phase(Humble and Kim, 2018).

The process of validating the brainstormed idea is an experiment. An experiment begins with a clear hypothesis that makes predictions about what is supposed to happen. The hypothesis is a sentence used to represent a business outcome, that can be validated when feedback on performance is acquired. The goal of defining a hypothesis is to validate two important assumptions (Hart, 2012): ***a)*** The value hypothesis: test whether a product or service really delivers value to customers once using it. ***b)*** The growth hypothesis – test how new customers will discover a product or service.

5.4.4 Prototype

Prototyping is a technique used for rapidly turning an idea into a more tangible artifact. Prototypes help to create a visual conceptualization of the solution idea that the team is brainstorming.

The prototypes can be of different types. Paper mock-ups are the simplest type. Wireframes are a bit more sophisticated prototypes, that are created by using simple sketches and widgets to build a user interface prototype. The prototyping technique offers the advantage of building in a short time frame. In this activity, the designers, like UX, UI designers have the lead for bringing the idea into life.

5.4.5 Evaluate

The prototype evaluation activity serves as a very first feedback point with the user, which is fundamental for the BizDevOps process. For this phase, usually a small group of users is appointed. Feedback on the prototyped solution is gathered by interviews with user or by observing them performing tasks. After collecting all the feedback, the results are synthesized again. Now that more insights on the idea requirements have been identified, from the user testing, there might be a change in the point of view (PoV) created.

This activity is crucial to early identifying not value adding ideas, that would cost the organization effort and resources. Therefore, there is a decision making point on

whether to continue further with the same idea or to abandon it and iterate again on this phase, by going back to the Understand or Explore activity.

At the beginning when you don't have a product it is just guessing and prototype. But when you start building the product and growing, this person [UX] should test this product and get more feedback from the customers... It might happen that the new functionalities that we implemented based on the requirements are wrong and it is better to approach that in an early stage and fix it as soon as possible and don't wait until the end of the release. (iv5).

5.4.6 Plan and Commit

If the BizDevOps team thinks that the idea has the potential for increasing business value, the process continues to the next step of planning for the development of the idea.

The *Explore and Identify* phase is an eternal step, that enables the organization to continuously search for new innovations, stay competitive and maintain a healthy state of the Business and Product Backlog. Unlike current state of the art, that requires the organization to apply activities related to the *Explore and Identify* phase in a timeframe of 3 months or more, in BizDevOps, these activities are continuous and with a shorter occurrence time of 2 weeks, similar to a Scrum Sprint. This way, the organization can reduce the risk that the BizDevOps team is working with the wrong goals and requirements.

In user-centered agile software development, this step of the process follows the pattern of Cycle Zero, that represents a sprint for analysis and design activities, that takes place before the actual agile software development (Brhel et al., 2015).

In the agile world, long-term project planning and setting milestones on the long-run is not suitable, due to the rapidly changing market and customer needs (iv7). To cope with the challenges of this traditional project planning approach, the idea of having a Business Backlog in place is introduced, which serves as a documentation and communication artifact for the product ideas identified in the *Explore* activity.

A Business Backlog is used to gather these ideas that emerge from the business stakeholders, customer requests or the BizDevOps team. These ideas can have different nature, like new features or functionalities of the product, new product ideas. Everybody should be able to contribute with new elements to the Business backlog at any time, not only during special innovation events. The refinement of the Business Backlog is responsibility of the Business stakeholders, that evaluate which of the ideas on the backlog provide a higher business value. The Business Backlog refinement is happening continuously by collaborating with the PO. When an idea is picked up from the Business backlog, BizDevOps teams are created to continue with the execution of the idea.

Behaviour Driven Development (BDD) is a method for software development that aims to foster collaboration between developers, testers, QAs, business and Product Managers. BDD is a way of documenting requirements, by making use of scenarios that are written from an end-users perspective (Perera et al., 2017).

After an initial prototype evaluation with users, the prototype is turned into working software. A BizDevOps team is created and committed to further develop the MVP concept or prototype. The BizDevOps team works on splitting the MVP concept into functional and non-functional requirements, that are added to the Product Backlog, in the form of BDD specifications.

Exploration Demo is a ceremony that is used to facilitate the transition to the next phase of the BizDevOps process. This meeting is similar to the Sprint Demo in

a Sprint Review ceremony. In a Sprint Demo, the Scrum team provides a demonstration of the work that was done, in order to receive feedback from the different stakeholders and keep them informed about the progress. However, in this meeting, the roles are reversed. The PO gives a demo on what is expected for the upcoming Sprint. The Exploration Demo meeting helps creating a shared understanding of the product vision is created among all team participants. This is an important ceremony for the BizDevOps process, for enabling a close alignment.

The hard task of the PO prioritizing the product backlog items is facilitated in this event. With both the technical implementation view from developers and technical architects and the business priority view, the PO has the final decision of assigning priorities in the backlog items.

5.5 Phase two: Develop and Operate

In this phase the actual development of the product increment takes place. The decision to scale the prototype idea from the previous phase is made and a BizDevOps team is committed for the further development. The input to this phase is the prototypes idea and the list of functional and non-functional requirements.

As mentioned in the Process preconditions, the organization applying the BizDevOps process, is already familiar with the DevOps way of working. Therefore, we briefly discuss the activities in this phase.

In this phase, the Scrum team and the Operations function has a dominant role, while the business roles are mostly supporting the process.

Figure5.7 presents the activities of the *Develop and Operate* phase.

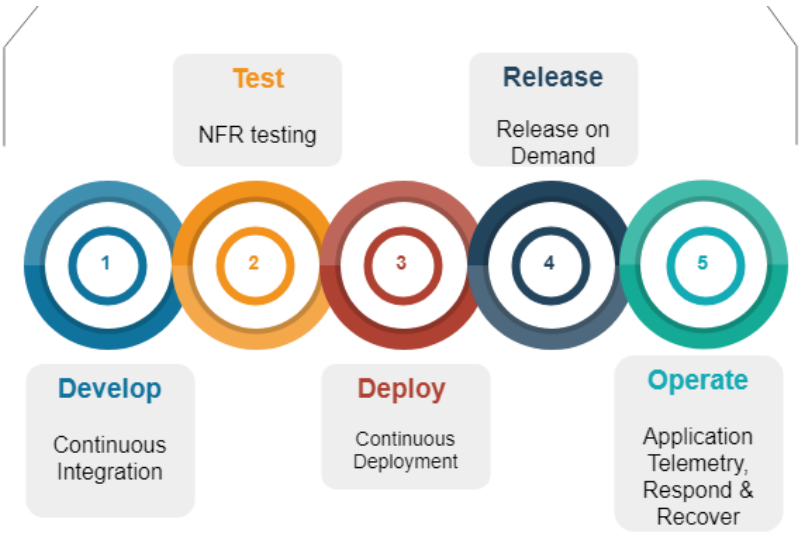

FIGURE 5.7: Phase 2: *Develop and Operate* activities

5.5.1 Develop

If this is the first iteration of this phase, the goal is to build an MVP. The work is divided in small batches as in Lean startup (Humble and Kim, 2018). The BizDevOps team works on scaling the created prototype. The MVP is a working version of the initial prototype, designed in the *Explore and Identify* phase. It contains the minimum necessary working features that can be used to obtain user feedback.

As mentioned before and as in agile methods, the activities are highly iterative. On the next iteration of this step, it is not an MVP anymore, but an existing product. The team is responsible for the software increment from end to end.

5.5.2 Test

In the Testing activity, beyond the standard types of automated tests that are performed in the testing environment, in BizDevOps, usability testing is an important enabler for maintaining the end user, as an important source of feedback. Usability testing is considered a non-functional requirement.

5.5.3 Deploy

In order to benefit from the BizDevOps process, it is proposed that the organization should be able to Continuously Deploy to production. Although, this is seen as a challenge to organizations that are currently still facing challenges with Continuous Delivery, it is argued that experimentation and innovation can benefit from the organization's capability to continuously deploy to production environment.

5.5.4 Release

In BizDevOps, the software release is decoupled from deployment and the term is known as Release on Demand.

At the moment that an increment is released, the organization might want to use innovative techniques for experimentation with the users, in order to measure the acceptance of a new feature that is made available in the release.

A/B testing is a technique used for experimentation. It is a form of statistical test, that is utilized in deployment of software. By applying A/B testing, the product increment is made available to two parallel production environments. The feature is manipulated systematically with the purpose to monitor user feedback. This is an effective way to identify value added features, by continuously monitoring user feedback (Fitzgerald and Stol, 2017) The origins of A/B testing come from Beta testing, which is used to elicit customer feedback on a product increment, prior to the release.

Other techniques are Canary deployments, Early access, Dark Launches. These forms of experimentation allow the BizDevOps team to ensure that the increment satisfies the user needs. The ability to continuously deploy to production is a facilitator. The BizDevOps team has the flexibility and authority to perform these experiments with real users. Having the business inside the team can remove the challenges of long waiting time for approval.

Being able to continuously deploy and building confidence in A/B testing or Blue-Green Release patterns allows the BizDevOps team to be more resilient towards rollbacks.

5.5.5 Operate

The last activity of this phase is operation of the software increment. Once the work is "done", the product increment is made available to the users and the operation, maintenance and monitoring of the system takes place.

Based on the performance checks, the overall performance can be assessed and in case of delivery blockers, the process can go back to the Exploration Sprint. During

the delivery lifecycle, the team might face bottlenecks, delivery blockers, technical issues, requirements prioritization issues. The decision to go back to the Explore and Identify phase may come as a result of poor team performance, unreached team goals or confusion on the quality or priority of user stories. Going back to the Explore and Identify phase can help the team to early identify bottleneck points and respond to them fast and with flexibility.

Pivot is a fundamental concept in Lean startup that indicates a structured change of course to test new hypotheses about the product. The decision to pivot comes after the execution of at least one build measure learn feedback loop, meaning that customer feedback is obtained and based on that, decisions can be made. A pivot can be thought as the decision to let go on the current hypothesis and look for an opportunity to grow, by identifying new business opportunities. A pivot decision requires the definition of new hypothesis and the creation of a new MVP.

BizDevOps requires a change in the way the software increment is operated. Deploying robust systems, that enable timely recovery to failure and can handle high operational load is not enough in BizDevOps. A key success point is to operate in a user-centered way. This indicates that software operations should be driven by the motive to help users achieve their goals, in order to maximize quality and value. In other words, as Lean UX emphasized, the process should be driven by the outcomes that the running software increment brings to the users. (Gothelf and Seiden, 2016).

5.6 Phase Three: Validation

This phase serves as an end-to-end performance checkpoint. The main goal of this phase is to validate the hypothesis made in the *Explore and Identify* phase towards quantitative metrics and outcomes generated. The main purpose of this phase is to close the delivery cycle, by providing a point where the business can evaluate the whole product development effort.

Metrics are generated throughout the whole delivery cycle by making use of an integrated DevOps toolchain. In BizDevOps, the organization is data-driven. This is a good way to baseline improvement changes and validate your experimentation hypotheses.

Figure 5.8 presents the activities that take place in this phase.

5.6.1 Measure Business Value

Business value can be assessed by measuring and analyzing different metrics generated in the delivery cycle. Heidenberg et al. (2012) propose a model for measuring business value in lean and large-scale agile organizations. The authors propose six attributes that contribute to assessing business value, that can be utilized in BizDevOps: ***a)*** Monetary Value relates to the economical benefit realized from a feature; ***b)*** Market Enabler, when a feature is considered as facilitator of new market possibilities; ***c)*** Technical Enabler, when a feature works as foundation for future functionalities; ***d)*** Competence Growth, when developing features that might be subject to an initial learning curve in the team, but are profitable for the future; ***e)*** Employee Satisfaction, relates to the health of the team; ***f)*** Customer Satisfaction, is an important business metric.

FIGURE 5.8: Phase3: *Validate* activities

5.6.2 Validate

The BizDevOps team holds a Performance session. This meeting is different from the Sprint Review, because the increment is now evaluated towards the target KPIs that were set in the initial phase. This Performance session serves as the validation of the product increment.

You could have a performance session, like you have planning and retrospective sessions, in which you evaluate data from external input. This is your agile thermometer, to see if you need to be steered in another direction or not. In this session you can take a look at your KPIs, can have external input from customers and everything you need to know to keep your software relevant, fitting. (iv10)

The BizDevOps process makes use of the Lean UX principle, *Focus on outcomes, rather than outputs* (Gothelf and Seiden, 2016). Outcomes are value-oriented and they should be the main drivers for performance checkpoints. With outputs, the impact on business value cannot be determined, therefore they do no provide a good measure for business value. For example, the output of the business and development sprint effort is the release of a new feature. It is hard to determine the business value that this new feature brought, ie. will this new feature encourage users to buy more from the website. However, by focusing on the outcome, the BizDevOps team is continuously exploring until they find a way that maximizes the business value.

Necessary input for the Performance session are operational data in the form of dashboards, market research is conducted. The organizations is getting insights from sales, Customer Relationship Management systems. The Performance Review session has as output a decision on whether the target business goals were reached, or the product idea is not bringing any value to the business anymore.

The Validation phase is focused on evaluating the outcomes, in order to assess the realized business value. To achieve that, it is important for the BizDevOps team to move from looking at isolated pieces of information, metrics or performance data.

But rather, focus on the overall effort of the team. For example, instead of being satisfied with having a happy state where the operations are able to respond fast to resolving open tickets by customers, the team should jointly evaluate more insightful metrics like, customer satisfaction or retention rate.

"You could have a performance session, like you have planning and retrospective sessions, in which you evaluate data from external input. This is your agile thermometer, to see if you need to be steered in another direction or not. In this session you can take a look at your KPIs, can have external input from customers and everything you need to know to keep your software relevant, fitting." (iv10).

Application Telemetry refers to the continuous tracking of the health and performance of the running application. The telemetry data originate from logs and metrics. Telemetry can offer insights on features, bugs and issues and offer better performance visibility for the organization (iv10).

Blameless Performance Review is an important attribute of the session. In innovation and experimentation, the teams have the responsibility to take risks and rapidly adapt to changing needs. Therefore, each individual of the team is equally responsible for a fail of a success.

5.6.3 Learn and Adapt

The Validation phase ends with learning from the past process and making decisions on how to respond. The learning process takes part in three levels. The *team level* implies the internal learning process that is happening during software development. IT refers to the Scrum team lessons learned and how the team establishes best practices. New knowledge that is acquired by the team (for example, new technologies) is identified. In the *delivery level* the BizDevOps team gathers learning on how the customers is perceiving new features and how new features impact the infrastructure. In the *business idea level* the BizDevOps team acquires knowledge on the realization of the expected impact of the idea. While this is of main concern for the business roles, all the other BizDevOps team members should be involved, in order to create a better understanding on the business value realized from the whole software delivery effort.

5.7 Chapter summary

In this chapter the *Treatment Design* phase of the Design Science cycle has been explained. This phase resulted in the design of the artifact, the BizDevOps process model, which is the primary contribution of this thesis. A description of the BizDevOps team is provided and it is an essential enabler for the successful application of the BizDevOps process model.

Chapter 6

Artifact Validation

This chapter describes the *Treatment Validation* phase. Validation of a treatment implies justifying that the implementation of the treatment in a problem context, would contribute to the realization of stakeholder goals (Wieringa, 2014). The results from the treatment validation has been reported and later, an improved artifact has been designed.

6.1 Evaluation results

The focus group validation session serves as a first point of evaluating the designed BizDevOps process model in a real world context. The focus group results are reported in this section. The results are discussed according to each of the evaluation criteria, namely *completeness*, *efficacy*, *fit with the organization* and *fit with people*.

6.1.1 Completeness

Regarding completeness, the BizDevOps process model was evaluated as partially complete. The participants were able to understand it and they found the visual representation clear. Although the model doesn't claim for completeness, participants expressed that it covers to a big extend what is seen in practice (participant2).

One major point of improvement is the missing actors in the model. Experts found it hard to grasp the roles involved in each phase of the process. All the participants expressed that they would like to see the actors in the model, which would cover the BizDevOps team organization, proposed as part of the artifact. Presenting the roles and responsibilities distribution with actors, that are involved through the whole process, but with different amount of dedication can help convey the BizDevOps team organization (participant1).

Another point concerns the naming of phase three, *Validation*, as it was not found very intuitive by the experts. The name *Value delivery validation* or *Business value validation* was suggested instead, in order to specify that in this phase a business evaluation of the whole software delivery effort is performed.

The first phase of the BizDevOps process model emphasizes an innovative approach for the continuous elicitation of new product requirements. However, the origin of requirements is not always the business or end user side. There can be some pressure from the technology side to give priority to technological aspects (participant2). The model should take into account not only the end user requirements, but also the technical enablers and non-functional requirements that are an inevitable part of software maintenance.

At last, the process model does not show where the architecture solution part of the product idea is generated (participant2, participant4). It is a common practice

now to do things by design and since the BizDevOps model is introducing a high-paced way of working, it is important that many parts of the product idea, such as non-functional requirements are covered and checked early on the process. participant1 suggested in the *Prototype* activity.

6.1.2 Efficacy

The artifact was evaluated as effective in serving the intended goal. The BizDevOps process model was perceived as a value enabler, when it is applied on top of an existing DevOps framework. When applying the BizDevOps process model, the business stakeholders will feel empowered, by being more involved in the whole delivery cycle and this has a positive impact in achieving better results (participant2). Therefore, the introduction of the BizDevOps team was perceived as a good fit for the purpose of the designed artifact.

In addition, it was noted that the BizDevOps model will accelerate the organizations ability to respond to external changes. The business always wants to be able to "pull the plug" whenever they see that the company is steered to the wrong direction. By continuously and actively involving the business stakeholders throughout the process, this objective can be reached in an even faster pace. However, to avoid falling back to the old scenario of long waiting times for approvals, this process has to be optimized or regulated (participant4).

The BizDevOps process model would accelerate the digital transformation for organizations that are enthusiastic about change, would stimulate innovation and would build a strong baseline to being a digital organization (participant2).

6.1.3 Fit with the organization

Regarding the suitability of the artifact when adopted by an organization, the participants stated several point of concern.

The organization should be of a certain fit to be able to apply the designed artifact (participant4). Influencing factors that were identified in the validation session are:

1. Organization maturity: While being DevOps mature is stated as prerequisite for the application of the designed artifact, the experts raised the concern that DevOps is understood and adopted differently in different organizations and

2. Domain of operation: The BizDevOps process model was evaluated as not applicable in all sectors, due to the regulatory constraints that exist in several sectors. The financial sector was mentioned as an example. Safety, environmental and regulatory constraint apply most of the time in every industry and this process has to take these into account (participant4). The process would be applicable for organizations like Innovation Hubs and mature companies, that would be able to implement it with relatively low cost. But an organization in the current state of the market is difficult to implement that now (participant1, participant2).

3. Culture: is a key aspect that can either help an organization benefit from this model or can result in a failure (participant3). But in an ideal DevOps situation, if this model would be applied regardless of the cultural constraints, it would further build upon having a culture of trust in the whole pipeline.

4. Roles and budget: The business will be concerned about the costs of assigning new roles in the team and if their involvement is really bringing value to

the business (participant3). This sort of resistance is understandable, since the BizDevOps model would go through different stages of acceptance, until its maturity.

A strong point associated with the artifact is that it promotes a flat organizational structure, resulting in the main benefit of having faster feedback between the different roles. This enables the organization to be high-paced and responsive to change (participant2).

6.1.4 Fit with people

The evaluation of the artifact regarding the fit with people, was centered around the acceptance of the BizDevOps team, that is introduced in the designed artifact. The BizDevOps team follows the principles of DevOps teams, being cross-functional and continuously working together for improving the design, architecture, code, deploy and operate the software increment (participant3). Having everyone as part of the process from beginning to end, but contributing with different amount of effort during the whole delivery lifecycle is a good solution (participant1). An alternative approach suggested was adding the category of *Enabler* roles, as a group of actors, that support the BizDevOps team in the delivery. In this group of *Enabler* roles, the managerial levels can be included, or the technical/solution architects, that are not dedicated roles of the BizDevOps team.

6.2 Artifact evolution after validation

The results of the validation session yell for several improvements of the artifact. It is however, relevant to emphasize the boundaries of this research and to what extend the improvement points can be considered in our scope.

Several external factors can influence the successful adoption of the designed artifact, as mentioned by the experts, the organizations maturity level, the culture, the domain of operation. However, this research proposes an abstract model for the application of BizDevOps, which leaves enough room for organizations to tailor the process to their specific needs. Therefore, the influence of these external factors cannot be represented in an abstract model, but it is rather left as part of the evaluation discussion. Therefore, the only changes that can be depicted in the BizDevOps process model are presented below.

- **Adding Actors in the model** Since the BizDevOps process model puts a great emphasis on team re-organization, adding actors in the model would make it more intuitive to convey the message of the multidisciplinary team.
- **Renaming the *Validate* phase** The name of the third phase has been changed to *Validate Business Value* to emphasize that this is a business evaluation step and is performed in an organization wide level.

Figure 6.1 presents the improved version of the BizDevOps process model.

6.3 Chapter summary

In this chapter, the findings from the Treatment Validation phase have been reported. Based on the current research context and scope, an improved version of the artifact

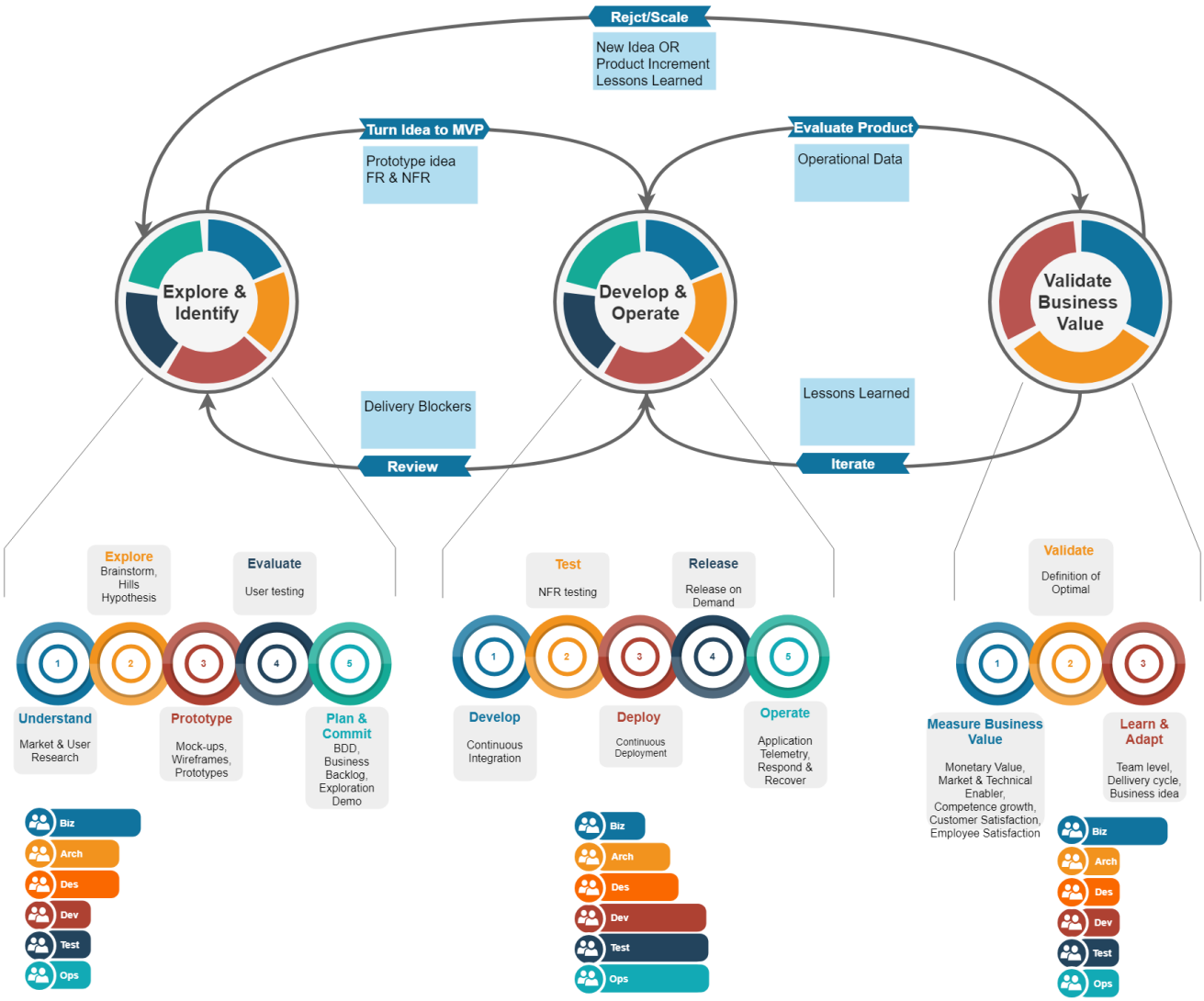

FIGURE 6.1: The BizDevOps process model after validation

is designed. The rational behind these improvements is explained, along with an argumentation of how the improvements would contribute to a better artifact design.

The main conclusion from the artifact validation is that the BizDevOps process model is valid and partially complete. The model provides a good baseline for organizations that seek for innovation and become high paced.

Chapter 7

Discussion and limitations

7.1 Discussion

In this research, a BizDevOps process model has been designed. The process model is an abstract representation of the delivery lifecycle and aims to provide agile organizations with a generic picture of how to maximize value in delivering software, by integrating the capabilities of continuously identifying product ideas that bring value to the business, developing and operating software products and evaluating the outcomes of the delivery effort. The above mentioned integration enables a closed loop of activities starting from business to development and operation of the software product, that can contribute to a successful goal alignment.

This study is comparable to other studies, which contribute to the definition of BizDevOps and the integration of user-centered approaches with agile software development. Gruhn and Schäfer (2015) focus on an architectural approach of BizDevOps and emphasize the end-user involvement in the process. However, this approach is lacking an explicit workflow-like description of how the business collaborates with the DevOps team. Forbrig and Herczeg (2015) discusses the integration of human-centered design with Scrum, but the roles description in the whole process is left an open question. The authors propose the possible existence of a BizDevOps team, in the same way as it is further described in our research.

In addition, Wiedemann et al. (2019) and Schrader and Droegehorn (2018) provide a good starting point towards the definition of BizDevOps teams. Our contribution adds to the mentioned studies a more detailed description of how the BizDevOps team would interact in the different activities of software delivery.

7.2 Scientific and societal contribution

This research contributes to the current scientific body of knowledge with an innovative proposal for software delivery activities, suitable for agile, end-user oriented organizations. Moreover, this is one of the first studies to give an abstract, yet workflow oriented approach to the definition of BizDevOps.

The proposed BizDevOps model provides a bridge between requirements elicitation activities for product design with the software delivery cycle and serves as a baseline for what mechanisms should be in place, in order to support agile organizations with becoming more innovative and eliminating siloed culture. The high level of abstraction allows organizations for tailoring the process model to their specific needs.

The societal contribution of this research highlights the importance of establishing new organizational models, like the BizDevOps model (including the organization in BizDevOps teams), in order to keep a fast pace and stay competitive in

the current rapidly changing environment. Furthermore, as evaluated by industry experts, the BizDevOps process model supports agile organization's movement towards becoming digital and innovative.

7.3 Limitations

In this section, the limitations of this study are discussed, in order to provide the reader with the validity and trustworthiness of the results presented in Chapter 5 and the research approach followed, as explained in Chapter 2. The classification of validity threats has been adapted from Yin (2009).

Construct Validity

Construct Validity is concerned with the extend to which the operational measures being used in the research are valid. To increase the credibility and validity of the results, triangulation has been used. *Data triangulation* is performed by using multiple sources of evidence, namely scientific literature, interviews and a focus group validation with experts. Second, the interviewees sample was representative, as different categories of views were included. Specifically, to mitigate the biased opinion of only experts from the consulting sector, external interviews were conducted. Experts from outside the Netherlands, representing the view of software product companies working with internal DevOps teams; Vendors providing technology solutions for a DevOps infrastructure; and the voice of the research community.

Performing literature review by means of the systematic mapping technique brings the limitation of not being able to study the research area in depth. Due to the fact that mapping studies have as a primary goal to structure the research area, the sample of publications are not reviewed in a systematic way, as in Systematic review (Petersen et al., 2008; Petersen et al., 2015). As a result, some concepts might have been overlooked, because they did not appear in the review of the sample of papers. Moreover, the inclusion of *traceability* concept in the keywords for the literature search and later excluding the concept for the artifact design, might have influenced the inference of results.

External Validity

External Validity is concerned with the extend to which the results obtained in the current study can be generalized and can be used in different context, people or place (Wohlin et al., 2012). This research cannot claim for generalization of results, due to the fact that the research was influenced by the context of the company, that was one of the stakeholders of this research. One measure that has been taken to reduce this validity threat was conducting external interviews. However, seven out of ten interviews and the focus group participants were Accenture employees. Even though the research participants have been asked to talk about their experience with different clients, there is a high chance that the views and opinions have been affected by the company culture and other internal factors.

Reliability

Reliability is concerned with the extent to which, the data and the analysis performed are dependent on the researcher and if the same results would be obtained, if the study would be conducted by another researcher (Wohlin et al., 2012). The

measures taken to mitigate reliability issues include systematic documentation of all the steps performed in this research and following scientific approaches and predefined protocols for the literature search and interviews (presented in Appendices A, B and C), as well as defining the operationalization of the evaluation criteria prior to the focus group validation session (described in Section 2.4.1).

Chapter 8

Conclusions

In this chapter, the main conclusions of the research are summarized. The answers to the research questions are discussed, a summary of the main contributions of the entire research process is provided and potential future work directions are proposed.

8.1 Answer to Research Questions

RQ1: "What is the motivation of organizations to have continuous business alignment in their DevOps process?"

This research question is answered by conducting a literature review, using the Systematic mapping technique and expert interviews. DevOps is perceived as a mindset, that enables organizations to achieve faster and more quality delivery. Among the reasons why agile organizations would adopt DevOps is becoming innovative and more competitive to the market. However, the realization of these goals is hindered by several alignment challenges. Often, a big resistance from the business side to adapt to this fast pace of working is observed in practice. This makes the business involvement in the software delivery cycle to be seen as a barrier for fast delivery, as advocated by DevOps. As a result, there is still a barrier, which causes a misalignment in goals and objectives of business and DevOps teams. Organizations that are already experienced with agile and DevOps way of working recognize this as a barrier and support that the full potential of adopting DevOps can be realized only when this barrier is eliminated. The findings of this question are reported in Chapter 3 and 4.

RQ2:"How can we extend the DevOps process, in order to achieve continuous business alignment? "

This research question is answered by conducting a second round of expert interviews and using the knowledge from related works. Based on the list of requirements identified from the first research question, the BizDevOps process model is designed, that describes the high-level activities that should be executed in an iterative way. The process model consists of three main phases, that describe the end-to-end delivery process. The BizDevOps process model is the primary artifact of this research, however its successful application goes hand in hand with the understanding of the BizDevOps team concept and the acceptance of this way of working. The designed artifact is explained in Chapter 5.

RQ3:"How is the BizDevOps process model evaluated by experts?"

This research question aims to evaluate the designed artifact and identify improvement points. A focus group session has been conducted, where experts have evaluated the completeness, efficacy and how the BizDevOps model would fit with

the organization and people. According to the experts opinion, the artifact has been perceived as understandable and a correct representation of what is seen in practice. A primary benefit of applying the BizDevOps process model is the making the organization very high-paced and responsive to change, the business stakeholders being empowered and as a result a better stakeholders alignment in software delivery can be achieved. Several improvements are applied to the initial artifact, like adding actors in the model to better convey the message of the BizDevOps team and renaming some process components for better understandability. The findings of this research questions are described in Chapter 6.

MRQ:"How to design a process for the continuous alignment of business goals with DevOps"

The main research question is answered by combining the answers of the above sub-questions. The designed artifact provides an abstract process model, that expands the end-to-end software delivery lifecycle and enables rapid software delivery. The concept of continuous alignment supports the current organization's needs to stay competitive and responsive to change. In the BizDevOps process model, the continuous alignment is supported by emphasizing the tight link between product design and product development, early user involvement in the process and frequent performance checkpoints. Also, by ensuring the commitment of business stakeholders in BizDevOps teams that: continuously iterate on the pre-development activities (Explore and Identify phase); facilitate the development process (Develop and Operate phase); and evaluate the delivery effort (Validate Business Value phase) can mitigate the risk of having unreached business goals.

Although the process model is evaluated as successfully achieving its goals, its applicability is still questioned, due to restrictions that my apply in different sectors.

8.2 Future Work

We believe this research area will gain more and more interest over the coming years. This study provides a very abstract description of the BizDevOps process model and a starting point for the agile organization that wants to achieve a close business and DevOps alignment. The future research directions are categorized below:

- **Demonstrating the applicability of the BizDevOps process model in a concrete use scenario.**

  Since the research area is very new, more validation and evaluation research would be a good approach, to assess and observe the application of the artifact in the real environment. Therefore, some interesting aspects to be observed in a case study application are:

  a) How the model would fit when applied in an existing DevOps framework. Would there be a need for specifying new principles as an addition to the known DevOps principles.

  b) Acceptance of the BizDevOps team. This aspect is closely related to human factors. Since the model proposes a change in the way of working, several constraints, as have been identified in this research may apply.

  c) Evaluation of the model application outcomes. It would be interesting to see if the real benefits of enhancing business value by enabling a closer

business alignment and end-user involvement are realized when applying the model. By looking at the real-world application of the process model, further points of improvement can be identified.

- **Formalizing the BizDevOps process model by designing a method.**

  The current process model presents a rather high level and abstract representation of the whole delivery cycle. Designing a method would provide a formal approach to software and systems development with the BizDevOps model. A method is defined as an approach that entails a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding products (Brinkkemper, 1996). We strongly believe that the BizDevOps process model proposes a new way of thinking for the organization and has the potential to be extended with structured activities and agreed deliverables between these activities.

- **Studying organization's Change Management towards the adoption of BizDevOps.**

  A current trend for organizations, as observed in practice, is becoming data-driven. We claim that the BizDevOps process model supports this movement for the agile organization and it is an interesting area for future research to provide a holistic BizDevOps framework.

8.3 Main contribution

This section concludes the thesis report. We provide a brief summary of the main contributions of the entire research process followed in this thesis. The contributions are highlighted below:

- First, the state of the art in scientific literature has been studied. By conducting a Systematic mapping research and an additional literature search, we were able to structure the research gap, understand the research domain and create an understanding of the main concepts of the research problem.
- Next, the industry perspective on the problem has been captured and together with knowledge from literature, a holistic understanding has been created.
- Later, the creation of the BizDevOps process model (and the BizDevOps team description) has been the primary contribution of this thesis.
- The artifact has been validated with practitioners and the acquired feedback has contributed to an improved version of it.
- At last, envisioning future work provides researchers with a starting point to further explore this research domain.

Bibliography

Abdelkebir, S., Maleh, Y., and Belaissaoui, M. (2017). *"An Agile Framework for ITS Management In Organizations: A Case Study Based on DevOps"*. In: *Proceedings of the 2Nd International Conference on Computing and Wireless Communication Systems*. ICCWCS'17. Larache, Morocco: ACM, 67:1–67:8. ISBN: 978-1-4503-5306-9. DOI: 10.1145/3167486.3167556. URL: http://doi.acm.org/10.1145/3167486.3167556.

Alqudah, M. and Razali, R. (2016). *"A review of scaling agile methods in large software development"*. In: *International Journal on Advanced Science, Engineering and Information Technology* 6.6, pp. 828–837. DOI: http://dx.doi.org/10.18517/ijaseit.6.6.1374.

Babar, Z., Lapouchnian, A., and Yu, E. (2015). *"Modeling DevOps Deployment Choices Using Process Architecture Design Dimensions"*. In: *The Practice of Enterprise Modeling*. Ed. by J. Ralyté, S. España, and Ó. Pastor. Cham: Springer International Publishing, pp. 322–337. ISBN: 978-3-319-25897-3.

Balalaie, A., Heydarnoori, A., and Jamshidi, P. (2016). *"Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture"*. In: *IEEE Software* 33.3, pp. 42–52. ISSN: 0740-7459. DOI: 10.1109/MS.2016.64.

Benguria, G., Alonso, J., Etxaniz, I., Orue-Echevarria, L., and Escalante, M. (2018). *"Agile Development and Operation of Complex Systems in Multi-technology and Multi-company Environments: Following a DevOps Approach"*. In: *European Conference on Software Process Improvement*. Springer, pp. 15–27.

Blueprint, S. (2017). *Business Alignment Through the DevOps Loop*. Tech. rep. Unified Compliance. URL: https://www.unifiedcompliance.com/wp-content/uploads/2017/04/Whitepaper-StorytellerBizDevOps.pdf.

Bock, A., Kattenstroth, H., and Overbeek, S. (2014). *"Towards a Modeling Method for Supporting the Management of Organizational Decision Processes"*. In: *Modellierung 2014*. Ed. by H.-G. Fill, D. Karagiannis, and U. Reimer. Bonn: Gesellschaft für Informatik e.V., pp. 49–64.

Braun, V. and Clarke, V. (2006). *"Using thematic analysis in psychology"*. In: *Qualitative Research in Psychology* 3.2, pp. 77–101. DOI: 10.1191/1478088706qp063oa. URL: https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa.

Brhel, M., Meth, H., Maedche, A., and Werder, K. (2015). *"Exploring principles of user-centered agile software development: A literature review"*. In: *Information and Software Technology* 61, pp. 163 –181. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2015.01.004. URL: http://www.sciencedirect.com/science/article/pii/S0950584915000129.

Brinkkemper, S. (1996). *"Method engineering: engineering of information systems development methods and tools"*. In: *Information and Software Technology* 38.4. Method Engineering and Meta-Modelling, pp. 275 –280. ISSN: 0950-5849. DOI: https://doi.org/10.1016/0950-5849(95)01059-9. URL: http://www.sciencedirect.com/science/article/pii/0950584995010599.

Carell, A., Lauenroth, K., and Platz, D. (2018). *"Using Design Thinking for Requirements Engineering in the Context of Digitalization and Digital Transformation: A Motivation and an Experience Report"*. In: *The Essence of Software Engineering*. Ed. by V. Gruhn and R. Striemer. Cham: Springer International Publishing, pp. 107–120. ISBN: 978-3-319-73897-0. DOI: 10.1007/978-3-319-73897-0_7. URL: https://doi.org/10.1007/978-3-319-73897-0_7.

Carniel, C. A. and Pegoraro, R. A. (2018). *"Metamodel for Requirements Traceability and Impact Analysis on Agile Methods"*. In: *Agile Methods*. Ed. by V. A. d. Santos, G. H. L. Pinto, and A. G. Serra Seca Neto. Cham: Springer International Publishing, pp. 105–117. ISBN: 978-3-319-73673-0.

Cleland-Huang, J. (2012). *"Traceability in Agile Projects"*. In: *Software and Systems Traceability*. Ed. by J. Cleland-Huang, O. Gotel, and A. Zisman. London: Springer London, pp. 265–275. ISBN: 978-1-4471-2239-5. DOI: 10.1007/978-1-4471-2239-5_12. URL: https://doi.org/10.1007/978-1-4471-2239-5_12.

Cleland-Huang, J., Rahimi, M., and Mäder, P. (2014). *"Achieving Lightweight Trustworthy Traceability"*. In: *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. FSE 2014. Hong Kong, China: ACM, pp. 849–852. ISBN: 978-1-4503-3056-5. DOI: 10.1145/2635868.2666612. URL: http://doi.acm.org/10.1145/2635868.2666612.

Cleland-Huang, J., Gotel, O. C. Z., Huffman Hayes, J., Mäder, P., and Zisman, A. (2014). *"Software Traceability: Trends and Future Directions"*. In: *Proceedings of the on Future of Software Engineering*. FOSE 2014. Hyderabad, India: ACM, pp. 55–69. ISBN: 978-1-4503-2865-4. DOI: 10.1145/2593882.2593891. URL: http://doi.acm.org/10.1145/2593882.2593891.

Di Nitto, E., Jamshidi, P., Guerriero, M., Spais, I., and Tamburri, D. A. (2016). *"A Software Architecture Framework for Quality-aware DevOps"*. In: *Proceedings of the 2Nd International Workshop on Quality-Aware DevOps*. QUDOS 2016. Saarbrücken, Germany: ACM, pp. 12–17. ISBN: 978-1-4503-4411-1. DOI: 10.1145/2945408.2945411. URL: http://doi.acm.org/10.1145/2945408.2945411.

Diebold, J., Diebold, P., and Vetter, A. (2018). *"Agile Meets Assessments: Case Study on How to Do Agile Process Improvement in a Very Small Enterprise"*. In: *Product-Focused Software Process Improvement*. Ed. by M. Kuhrmann et al. Cham: Springer International Publishing, pp. 31–47. ISBN: 978-3-030-03673-7.

Dittrich, Y., Nørbjerg, J., Tell, P., and Bendix, L. (2018). *"Researching Cooperation and Communication in Continuous Software Engineering"*. In: *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering*. CHASE '18. Gothenburg, Sweden: ACM, pp. 87–90. ISBN: 978-1-4503-5725-8. DOI: 10.1145/3195836.3195856. URL: http://doi.acm.org/10.1145/3195836.3195856.

Dobrigkeit, F., Paula, D. de, and Uflacker, M. (2019). *"InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup"*. In: *Design Thinking Research : Looking Further: Design Thinking Beyond Solution-Fixation*. Ed. by C. Meinel and L. Leifer. Cham: Springer International Publishing, pp. 199–227. ISBN: 978-3-319-97082-0. DOI: 10.1007/978-3-319-97082-0_11. URL: https://doi.org/10.1007/978-3-319-97082-0_11.

Dzvonyar, D., Krusche, S., Alkadhi, R., and Bruegge, B. (2016). *"Context-aware User Feedback in Continuous Software Evolution"*. In: *Proceedings of the International Workshop on Continuous Software Evolution and Delivery*. CSED '16. Austin, Texas: ACM, pp. 12–18. ISBN: 978-1-4503-4157-8. DOI: 10.1145/2896941.2896952. URL: http://doi.acm.org/10.1145/2896941.2896952.

Elberzhager, F., Holl, K., Karn, B., and Immich, T. (2017). *"Rapid Lean UX Development Through User Feedback Revelation"*. In: *Product-Focused Software Process Improvement*. Ed. by M. Felderer et al. Cham: Springer International Publishing, pp. 535–542. ISBN: 978-3-319-69926-4.

Fabijan, A., Dmitriev, P., Olsson, H. H., and Bosch, J. (2017). *"The Evolution of Continuous Experimentation in Software Product Development: From Data to a Data-driven Organization at Scale"*. In: *Proceedings of the 39th International Conference on Software Engineering*. ICSE '17. Buenos Aires, Argentina: IEEE Press, pp. 770–780. ISBN: 978-1-5386-3868-2. DOI: 10.1109/ICSE.2017.76. URL: https://doi.org/10.1109/ICSE.2017.76.

Farooqui, S. M. (2018). *"Accelerating Towards DevOps"*. In: *Enterprise DevOps Framework: Transforming IT Operations*. Berkeley, CA: Apress, pp. 95–105. ISBN: 978-1-4842-3612-3. DOI: 10.1007/978-1-4842-3612-3_9. URL: https://doi.org/10.1007/978-1-4842-3612-3_9.

Feijter, R. de, Overbeek, S., Vliet, R. van, Jagroep, E., and Brinkkemper, S. (2018). *"DevOps Competences and Maturity for Software Producing Organizations"*. In: *Enterprise, Business-Process and Information Systems Modeling*. Ed. by J. Gulden et al. Cham: Springer International Publishing, pp. 244–259. ISBN: 978-3-319-91704-7.

Ferre, X., Juristo, N., and Moreno, A. (2005). *"Obstacles for the integration of HCI practices into software engineering development processes"*. In: pp. 422–428. DOI: 10.4018/978-1-59140-562-7.ch064.

Fitzgerald, B. and Stol, K.-J. (2014). *"Continuous Software Engineering and Beyond: Trends and Challenges"*. In: *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*. RCoSE 2014. Hyderabad, India: ACM, pp. 1–9. ISBN: 978-1-4503-2856-2. DOI: 10.1145/2593812.2593813. URL: http://doi.acm.org/10.1145/2593812.2593813.

Fitzgerald, B. and Stol, K.-J. (2017). *"Continuous software engineering: A roadmap and agenda"*. In: *Journal of Systems and Software* 123, pp. 176 –189. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2015.06.063. URL: http://www.sciencedirect.com/science/article/pii/S0164121215001430.

Fontana, R. M., Fontana, I. M., Rosa Garbuio, P. A. da, Reinehr, S., and Malucelli, A. (2014). *"Processes versus people: How should agile software development maturity be defined?"* In: *Journal of Systems and Software* 97, pp. 140 –155. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2014.07.030. URL: http://www.sciencedirect.com/science/article/pii/S0164121214001587.

Forbrig, P. (2016a). *"Continuous Software Engineering with Special Emphasis on Continuous Business-Process Modeling and Human-Centered Design"*. In: *Proceedings of the 8th International Conference on Subject-oriented Business Process Management*. S-BPM '16. Erlangen, Germany: ACM, 11:1–11:4. ISBN: 978-1-4503-4071-7. DOI: 10.1145/2882879.2882895. URL: http://doi.acm.org/10.1145/2882879.2882895.

Forbrig, P. (2016b). *"When Do Projects End? – The Role of Continuous Software Engineering"*. In: *Perspectives in Business Informatics Research*. Ed. by V. Řepa and T. Bruckner. Cham: Springer International Publishing, pp. 107–121. ISBN: 978-3-319-45321-7.

Forbrig, P. (2016c). *"When Do Projects End? – The Role of Continuous Software Engineering"*. In: *Perspectives in Business Informatics Research*. Ed. by V. Řepa and T. Bruckner. Cham: Springer International Publishing, pp. 107–121. ISBN: 978-3-319-45321-7.

Forbrig, P. (2018a). *"BizDevOps and the Role of S-BPM"*. In: *Proceedings of the 10th International Conference on Subject-Oriented Business Process Management*. S-BPM One

'18. Linz, Austria: ACM, 1:1–1:8. ISBN: 978-1-4503-5360-1. DOI: 10.1145/3178248.3178250. URL: http://doi.acm.org/10.1145/3178248.3178250.

Forbrig, P. (2018b). *"Use Cases, User Stories and BizDevOps."* In: *REFSQ Workshops*.

Forbrig, P. and Dittmar, A. (2019). *"Integrating HCD into BizDevOps by Using the Subject-Oriented Approach"*. In: *Human-Centered Software Engineering*. Ed. by C. Bogdan, K. Kuusinen, M. K. Lárusdóttir, P. Palanque, and M. Winckler. Cham: Springer International Publishing, pp. 327–334. ISBN: 978-3-030-05909-5.

Forbrig, P. and Herczeg, M. (2015). *"Managing the Agile process of human-centred design and software development"*. In: *INTERACT*, pp. 223–232.

Freedman, R. (2016). *"The Agile Enterprise"*. In: *The Agile Consultant: Guiding Clients to Enterprise Agility*. Berkeley, CA: Apress, pp. 155–165. ISBN: 978-1-4302-6053-0. DOI: 10.1007/978-1-4302-6053-0_12. URL: https://doi.org/10.1007/978-1-4302-6053-0_12.

Goldkuhl, G., Lind, M., and Seigerroth, U. (1998). *"Method integration: the need for a learning perspective"*. In: *IEE Proceedings-Software* 145.4, pp. 113–118.

Gothelf, J. and Seiden, J. (2016). *Lean UX: designing great products with agile teams*. " O'Reilly Media, Inc."

Gruhn, V. and Schäfer, C. (2015). *"BizDevOps: Because DevOps is Not the End of the Story"*. In: *Intelligent Software Methodologies, Tools and Techniques*. Ed. by H. Fujita and G. Guizzi. Cham: Springer International Publishing, pp. 388–398. ISBN: 978-3-319-22689-7.

Häger, F. et al. (2015). *"DT@Scrum: Integrating Design Thinking with Software Development Processes"*. In: *Design Thinking Research: Building Innovators*. Ed. by H. Plattner, C. Meinel, and L. Leifer. Cham: Springer International Publishing, pp. 263–289. ISBN: 978-3-319-06823-7. DOI: 10.1007/978-3-319-06823-7_14. URL: https://doi.org/10.1007/978-3-319-06823-7_14.

Hall, M.-J. (2002). *"Aligning the Organization to Increase Performance Results. (Improving Organizational Performance)"*. In: *The Public Manager* 31.2, p. 7.

Hart, M. A. (2012). *"The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses Eric Ries. New York: Crown Business, 2011. 320 pages"*. In: *Journal of Product Innovation Management* 29.3, pp. 508–509.

Heidenberg, J., Weijola, M., Mikkonen, K., and Porres, I. (2012). *"A Model for Business Value in Large-Scale Agile and Lean Software Development"*. In: *Systems, Software and Services Process Improvement*. Ed. by D. Winkler, R. V. O'Connor, and R. Messnarz. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 49–60. ISBN: 978-3-642-31199-4.

Holtsnider, B., Wheeler, T., Stragand, G., and Gee, J. (2010). *"Overview of the six week solution"*. In: *Agile Development and Business Goals: The Six Week Solution*. Morgan Kaufmann.

Humble, J. and Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution.

Hung, R. Y.-Y., Chung, T., and Lien, B. Y.-H. (2007). *"Organizational Process Alignment and Dynamic Capabilities in High-Tech Industry"*. In: *Total Quality Management & Business Excellence* 18.9, pp. 1023–1034. DOI: 10.1080/14783360701594154. eprint: https://doi.org/10.1080/14783360701594154. URL: https://doi.org/10.1080/14783360701594154.

Jabbari, R., Ali, N., Petersen, K., and Tanveer, B. (2016). *"What is DevOps?: A Systematic Mapping Study on Definitions and Practices"*. In: pp. 1–11. DOI: 10.1145/2962695.2962707.

Jeong, S., Cho, H., and Lee, S. (2018). *"Agile Requirement Traceability Matrix"*. In: *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*. ICSE '18. Gothenburg, Sweden: ACM, pp. 187–188. ISBN: 978-1-4503-5663-3. DOI: 10.1145/3183440.3195089. URL: http://doi.acm.org/10.1145/3183440.3195089.

Jesse, N. (2018). *"Organizational Evolution - How Digital Disruption Enforces Organizational Agility"*. In: *IFAC-PapersOnLine* 51.30. 18th IFAC Conference on Technology, Culture and International Stability TECIS 2018, pp. 486 –491. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.11.310. URL: http://www.sciencedirect.com/science/article/pii/S2405896318329835.

Jones, S., Noppen, J., and Lettice, F. (2016). *"Management Challenges for DevOps Adoption Within UK SMEs"*. In: *Proceedings of the 2Nd International Workshop on Quality-Aware DevOps*. QUDOS 2016. Saarbrücken, Germany: ACM, pp. 7–11. ISBN: 978-1-4503-4411-1. DOI: 10.1145/2945408.2945410. URL: http://doi.acm.org/10.1145/2945408.2945410.

Kneuper, R. (2018). *"Foundations"*. In: *Software Processes and Life Cycle Models: An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes*. Cham: Springer International Publishing, pp. 1–39. ISBN: 978-3-319-98845-0. DOI: 10.1007/978-3-319-98845-0_1. URL: https://doi.org/10.1007/978-3-319-98845-0_1.

Krishna Kaiser, A. (2018). *"Introduction to DevOps"*. In: *Reinventing ITIL® in the Age of DevOps: Innovative Techniques to Make Processes Agile and Relevant*. Berkeley, CA: Apress, pp. 1–35. ISBN: 978-1-4842-3976-6. DOI: 10.1007/978-1-4842-3976-6_1. URL: https://doi.org/10.1007/978-1-4842-3976-6_1.

Kuranuki, Y., Ushio, T., Yasui, T., and Yamazaki, S. (2014). *"A New Business Model of Custom Software Development for Agile Software Development"*. In: *Proceedings of the International Workshop on Innovative Software Development Methodologies and Practices*. InnoSWDev 2014. Hong Kong, China: ACM, pp. 73–77. ISBN: 978-1-4503-3226-2. DOI: 10.1145/2666581.2666584. URL: http://doi.acm.org/10.1145/2666581.2666584.

Laukkarinen, T., Kuusinen, K., and Mikkonen, T. (2018). *"Regulated software meets DevOps"*. In: *Information and Software Technology* 97, pp. 176 –178. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2018.01.011. URL: http://www.sciencedirect.com/science/article/pii/S0950584918300144.

Lehtola, L., Kauppinen, M., Vähäniitty, J., and Komssi, M. (2009). *"Linking business and requirements engineering: Is solution planning a missing activity in software product companies?"* In: *Requir. Eng.* 14, pp. 113–128. DOI: 10.1007/s00766-009-0078-8.

Makki, M., Landuyt, D. V., Lagaisse, B., and Joosen, W. (2018). *"A comparative study of workflow customization strategies: Quality implications for multi-tenant SaaS"*. In: *Journal of Systems and Software* 144, pp. 423 –438. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2018.07.014. URL: http://www.sciencedirect.com/science/article/pii/S0164121218301420.

Mergel, I., Gong, Y., and Bertot, J. (2018). *"Agile government: Systematic literature review and future research"*. In: *Government Information Quarterly* 35.2. Agile Government and Adaptive Governance in the Public Sector, pp. 291 –298. ISSN: 0740-624X. DOI: https://doi.org/10.1016/j.giq.2018.04.003. URL: http://www.sciencedirect.com/science/article/pii/S0740624X18302107.

Muñoz, M. and Díaz, O. (2017). *"DevOps: Foundations and Its Utilization in Data Center"*. In: *Engineering and Management of Data Centers: An IT Service Management Approach*. Ed. by J. Marx Gómez, M. Mora, M. S. Raisinghani, W. Nebel, and R. V.

O'Connor. Cham: Springer International Publishing, pp. 205–225. ISBN: 978-3-319-65082-1. DOI: 10.1007/978-3-319-65082-1_10. URL: https://doi.org/10.1007/978-3-319-65082-1_10.

Nagarajan, A. D. and Overbeek, S. J. (2018). *"A DevOps Implementation Framework for Large Agile-Based Financial Organizations"*. In: *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*. Ed. by H. Panetto et al. Cham: Springer International Publishing, pp. 172–188. ISBN: 978-3-030-02610-3.

Olsson, H. H. (2018). *"Challenges and Strategies for Undertaking Continuous Experimentation to Embedded Systems: Industry and Research Perspectives"*. In: *Agile Processes in Software Engineering and Extreme Programming*, p. 277.

Palmer, K. D. (2014). *"The Essential Nature of Product Traceability and its Relation to Agile Approaches"*. In: *Procedia Computer Science* 28. 2014 Conference on Systems Engineering Research, pp. 44 –53. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2014.03.007. URL: http://www.sciencedirect.com/science/article/pii/S1877050914000702.

Patanakul, P. and Rufo-McCarron, R. (2018). *"Transitioning to agile software development: Lessons learned from a government-contracted program"*. In: *The Journal of High Technology Management Research* 29.2, pp. 181 –192. ISSN: 1047-8310. DOI: https://doi.org/10.1016/j.hitech.2018.10.002. URL: http://www.sciencedirect.com/science/article/pii/S1047831018300294.

Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., and Abrahamsson, P. (2014). *"Software development in startup companies: A systematic mapping study"*. In: *Information and Software Technology* 56.10, pp. 1200–1218.

Perera, P., Silva, R., and Perera, I. (2017). *"Improve software quality through practicing DevOps"*. In: pp. 1–6. DOI: 10.1109/ICTER.2017.8257807.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). *"Systematic Mapping Studies in Software Engineering."* In: *EASE*. Vol. 8, pp. 68–77.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). *"Guidelines for conducting systematic mapping studies in software engineering: An update"*. In: *Information and Software Technology* 64, pp. 1–18.

Pozzey, E., Wrigley, C., and Bucolo, S. (2012). *"Unpacking the opportunities for change within a family owned manufacturing SME: a design led innovation case study"*. In: *Leading innovation through design: Proceedings of the DMI 2012 international research conference*. DMI, pp. 841–855.

Prat, N., Comyn-Wattiau, I., and Akoka, J. (2014). *"Artifact Evaluation in Information Systems Design-Science Research-a Holistic View."* In: *PACIS*, p. 23.

Pérez-Castillo, R., Fernández-Ropero, M., and Piattini, M. (2019). *"Business process model refactoring applying IBUPROFEN. An industrial evaluation"*. In: *Journal of Systems and Software* 147, pp. 86 –103. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2018.10.012. URL: http://www.sciencedirect.com/science/article/pii/S016412121830222X.

Rahman, A., Agrawal, A., Krishna, R., and Sobran, A. (2018). *"Characterizing the Influence of Continuous Integration: Empirical Results from 250+ Open Source and Proprietary Projects"*. In: *Proceedings of the 4th ACM SIGSOFT International Workshop on Software Analytics*. SWAN 2018. Lake Buena Vista, FL, USA: ACM, pp. 8–14. ISBN: 978-1-4503-6056-2. DOI: 10.1145/3278142.3278149. URL: http://doi.acm.org/10.1145/3278142.3278149.

Ravichandran, A., Taylor, K., and Waterhouse, P. (2016). *"DevOps Finetuning"*. In: *DevOps for Digital Leaders: Reignite Business with a Modern DevOps-Enabled Software Factory*. Berkeley, CA: Apress, pp. 151–169. ISBN: 978-1-4842-1842-6. DOI:

10.1007/978-1-4842-1842-6_10. URL: https://doi.org/10.1007/978-1-4842-1842-6_10.

Rodríguez, P. et al. (2017). *"Continuous deployment of software intensive products and services: A systematic mapping study"*. In: *Journal of Systems and Software* 123, pp. 263–291.

Rong, G., Zhang, H., and Shao, D. (2016). *"CMMI Guided Process Improvement for DevOps Projects: An Exploratory Case Study"*. In: *Proceedings of the International Conference on Software and Systems Process*. ICSSP '16. Austin, Texas: ACM, pp. 76–85. ISBN: 978-1-4503-4188-2. DOI: 10.1145/2904354.2904372. URL: http://doi.acm.org/10.1145/2904354.2904372.

Satyal, S., Weber, I., Paik, H.-y., Di Ciccio, C., and Mendling, J. (2017). *"AB-BPM: Performance-Driven Instance Routing for Business Process Improvement"*. In: *Business Process Management*. Ed. by J. Carmona, G. Engels, and A. Kumar. Cham: Springer International Publishing, pp. 113–129. ISBN: 978-3-319-65000-5.

Schön, E.-M., Winter, D., Escalona, M. J., and Thomaschewski, J. (2017). *"Key challenges in agile requirements engineering"*. In: *International Conference on Agile Software Development*. Springer, Cham, pp. 37–51.

Schrader, I. and 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)*. The Steering Committee of The World Congress in Computer Science, Computer ..., pp. 87–93.

Schulte, S., Janiesch, C., Venugopal, S., Weber, I., and Hoenisch, P. (2015). *"Elastic Business Process Management: State of the art and open challenges for BPM in the cloud"*. In: *Future Generation Computer Systems* 46, pp. 36 –50. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2014.09.005. URL: http://www.sciencedirect.com/science/article/pii/S0167739X1400168X.

Scrum, L. S. (2018). *Feature Teams*. URL: https://less.works/less/structure/feature-teams.html (visited on 06/01/2019).

Senapathi, M. and Drury-Grogan, M. L. (2017). *"Refining a model for sustained usage of agile methodologies"*. In: *Journal of Systems and Software* 132, pp. 298 –316. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2017.07.010. URL: http://www.sciencedirect.com/science/article/pii/S0164121217301498.

Shahin, M., Babar, M. A., Zahedi, M., and Zhu, L. (2017). *"Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges"*. In: *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 111–120. DOI: 10.1109/ESEM.2017.18.

Sharma, S. (2017). *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise*. Wiley. ISBN: 9781119310761. URL: https://books.google.nl/books?id=jpfqDQAAQBAJ.

Shivakumar, S. K. (2018). *"Digital Transformation: A Project Management Case Study"*. In: *Complete Guide to Digital Project Management: From Pre-Sales to Post-Production*. Berkeley, CA: Apress, pp. 415–446. ISBN: 978-1-4842-3417-4. DOI: 10.1007/978-1-4842-3417-4_17. URL: https://doi.org/10.1007/978-1-4842-3417-4_17.

Smeds, J., Nybom, K., and Porres, I. (2015). *"DevOps: A Definition and Perceived Adoption Impediments"*. In: *Agile Processes in Software Engineering and Extreme Programming*. Ed. by C. Lassenius, T. Dingsøyr, and M. Paasivaara. Cham: Springer International Publishing, pp. 166–177. ISBN: 978-3-319-18612-2.

Ståhl, D., Hallén, K., and Bosch, J. (2016). *"Achieving traceability in large scale continuous integration and delivery deployment, usage and validation of the eiffel framework"*. In: *Empirical Software Engineering* 22, pp. 967–995.

Theunissen, T. and Van Heesch, U. (2017). *"Specification in Continuous Software Development"*. In: *Proceedings of the 22Nd European Conference on Pattern Languages of Programs*. EuroPLoP '17. Irsee, Germany: ACM, 5:1–5:19. ISBN: 978-1-4503-4848-5. DOI: 10.1145/3147704.3147709. URL: http://doi.acm.org/10.1145/3147704.3147709.

Top, Özden Özcan and Demirors, O. (2019). *"Application of a software agility assessment model – AgilityMod in the field"*. In: *Computer Standards Interfaces* 62, pp. 1 –16. ISSN: 0920-5489. DOI: https://doi.org/10.1016/j.csi.2018.07.002. URL: http://www.sciencedirect.com/science/article/pii/S0920548917303616.

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.

Unterkalmsteiner, M. et al. (2012). *"Evaluation and measurement of software process improvement—a systematic literature review"*. In: *IEEE Transactions on Software Engineering* 38.2, pp. 398–424.

Weiser, J. (2000). *"Organizational alignment: Are we heading in the same direction?"* In: *The Kansas Banker* 90, pp. 11–15.

Wettinger, J., Breitenbücher, U., Kopp, O., and Leymann, F. (2016). *"Streamlining DevOps automation for Cloud applications using TOSCA as standardized metamodel"*. In: *Future Generation Computer Systems* 56, pp. 317 –332. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2015.07.017. URL: http://www.sciencedirect.com/science/article/pii/S0167739X15002496.

Wiedemann, A., Wiesche, M., Gewald, H., and Krcmar, H. (2019). *"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. DOI: https://hdl.handle.net/10125/60138.

Wieringa, R., Maiden, N., Mead, N., and Rolland, C. (2005). *"Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion"*. In: *Requir. Eng.* 11.1, pp. 102–107. ISSN: 0947-3602. DOI: 10.1007/s00766-005-0021-6. URL: http://dx.doi.org/10.1007/s00766-005-0021-6.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer. ISBN: 978-3-662-43839-8. DOI: https://doi.org/10.1007/978-3-662-43839-8.

Wohlin, C. et al. (2012). *Experimentation in software engineering*. Springer Science & Business Media. ISBN: 978-3-642-29044-2. DOI: 10.1007/978-3-642-29044-2.

Wohlin, C. et al. (2013). *"On the reliability of mapping studies in software engineering"*. In: *Journal of Systems and Software* 86.10, pp. 2594 –2610. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2013.04.076. URL: http://www.sciencedirect.com/science/article/pii/S0164121213001234.

Yin, R. (2009). *"Case Study Research Design and Methods"*. In: 4th. Sage Publications, Beverly Hills.

Younas, M., Jawawi, D. N., Ghani, I., Fries, T., and Kazmi, R. (2018). *"Agile development in the cloud computing environment: A systematic review"*. In: *Information and Software Technology* 103, pp. 142 –158. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2018.06.014. URL: http://www.sciencedirect.com/science/article/pii/S0950584918301319.

Appendix A

Literature Review Protocol

The literature search is performed during December 2018 - January 2019.

A.1 Goal

The systematic mapping approach, followed in this study aims to:

1. explore and summarize the current state of the art on the relation between business processes and DevOps practices, artifact traceability in software engineering and agile practices
2. Identify research gaps
3. Identify trends in literature
4. Identify requirements for the artifact to be designed

A.2 Search Strategy

Perform the search of four keywords on each of the three databases and collect the first 10 resulting publications. All the collected publications go through the inclusion/exclusion criteria. Read abstracts of resulting papers and classify the according to: ***a)*** Research Type ***b)*** Contribution Type, by following the guidelines described in Section A.7. Perform thematic categorization of papers and determine the classification scheme.

A.3 Search Engines

The following search engines are used for this study:

- ACM (https://dl.acm.org)
- Science Direct (https://www.sciencedirect.com/)
- Springer (Link https://rd.springer.com/)

A.4 Keywords

- "DevOps" AND "business" OR "business process”
- "agile" OR "process" AND "DevOps"
- "traceability" AND "agile" OR "agile process“
- "large scale DevOps"

A.5 Inclusion criteria

Articles and papers that focus on the field of Software Engineering and Software Production, that appear as a result of the specified keywords search and are written in English.

A.6 Exclusion criteria

The following criteria has been used for exclusion of search results:

- papers outside the field of Software Engineering, Software Production
- older than 2014
- not written in English
- is a duplicate of another publication

A.7 Papers classification guide

TABLE A.1: Papers classification related to Research type adapted from (Wieringa et al., 2005)

Research type	**Description**
Validation Research	Techniques investigated are novel and have not yet been implemented in practice. Techniques used are for example experiments, i.e., work done in the lab.
Evaluation Research	Techniques are implemented in practice and an evaluation of the technique is conducted. That means, it is shown how the technique is implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benets and drawbacks (implementation evaluation). This also includes to identify problems in industry.
Solution Proposal	A solution for a problem is proposed, the solution can be either novel or a signicant extension of an existing technique. The potential benets and the applicability of the solution is shown by a small example or a good line of argumentation.
Philosophical Papers	These papers sketch a new way of looking at existing things by structuring the eld in form of a taxonomy or conceptual framework.
Opinion Papers	These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should been done. They do not rely on related work and research methodologies.
Experience Papers	Experience papers explain on what and how something has been done in practice. It has to be the personal experience of the author.

TABLE A.2: Papers classification related to Contribution type adapted from (Rodríguez et al., 2017)

Contribution type	**Description**
Model	Representation of an observed reality by concepts or related concepts after a conceptualization process.
Theory	Construct of cause-effect relationships of determined results.
Framework or method	Method or technique related to constructing software or managing development processes. Commonly it involves better ways to do some task.
Guidelines	List of advises, synthesis of the obtained research results.
Lessons learned	Set of outcomes, directly analyzed from the obtained research results.
Advice or implications	Discursive and generic recommendation, deemed from personal opinions.
Tools	Technology, programme or application used to create, debug, maintain or support software development processes.

Appendix B

Problem Investigation: Interview Protocol

Informed Consent

Exploratory study on DevOps practices and Business stakeholders alignment.


Utrecht University

accenture technology

This interview is conducted with regards to my master thesis research at Utrecht University. This research is part of an internship at Accenture. My research covers the topic of BizDevOps and how we can achieve better business stakeholders involvement into DevOps practices.

Through this interview, I am aiming to gain theoretical and practical knowledge about how particular DevOps activities take place and how the alignment with business goals is achieved. The main goal is to develop a process that guides organizations in applying a BizDevOps approach. The outcome of this research will be put available for use within Accenture.

All information gathered during the interview will be treated with respect and will only be used for scientific purposes. The interview will be recorded, transcribed and analyzed to draw scientific conclusions. All the information regarding people, companies and examples mentioned in the interview will remain confidential and they will be used only for the purpose of this scientific research. Entities mentioned in the interview will be anonymized to ensure confidentiality. The recording of this interview will be private, it will not be shared with other employees inside or outside Accenture, nor other organizations. The recordings will be permanently deleted after the research is completed and the concluded results will be used in my thesis.

Further, the interview does not aim to harm you, nor your organization. Therefore, you have all the right to stop the recording or the whole interview at any point, if you feel uncomfortable to continue. Participating in this interview is totally voluntary and only for supporting scientific purposes. Thank you for participating in this research.

If you have read and agree with the above statement, please sign below.

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

B.1 Interview Protocol

My name is Kleopatra Chasioti and I am following the Master in Business Informatics at Utrecht University. Currently, I am doing my master research project at Accenture. My research covers the topic BizDevOps and how we can have a better alignment between business goals and DevOps. This interview is designed for practitioners with experience in DevOps and software product delivery, either in academia, industry or both. The participants are required to answer these questions, based on their experience in the field. The main focus of this interview is to understand how agility is scaled in different organizational levels and what are current challenges or gaps with DevOps practices. Furthermore, the collaboration and communication mechanisms between different stakeholders are elicited, with the purpose to investigate their alignment.

The interview will take around 30 to 45 minutes. I would like to record this interview, for the purpose of further analysis. All the information will remain confidential.

May I start the recording?
*** Start Recording ***

B.1.1 Introduction

I would like to inform you that from now, our conversation is recorded.

1. Would you like to introduce yourself? What is your experience and expertise?
2. What does DevOps mean to you?

   Why would organizations adopt it?

   What benefits of adopting DevOps do you value most?
3. What exactly are your competencies? How are you involved in DevOps?

B.1.2 Scaling Agile

Scaling Agile in the whole organization is of a great interest, especially to large-scale organizations. The following questions ask how the Agile principles are scaled across different organizational levels, with a focus on DevOps way of working.

1. From your experience, how have you seen organizations deal with scaling Agile practices across the entire organization?
2. From your experience, how do organizations ensure that the Business goals and DevOps team(s) goals are aligned with each other?

   What roles are important to facilitate this alignment?
3. From your experience, do you think there is a gap between the Business unit and software development, in companies adopting DevOps?

   How is this alignment and collaboration currently enabled?

   Is this alignment happening in a top-down approach?

B.1.3 Business alignment in the DevOps lifecycle

With DevOps and breaking down silos between development and operations, organizations can empower DevOps teams and accelerate software delivery with shorter time-to-market. But, from the business point of view, time-to-value seems like a more important indicator. In this section, some questions will be asked with the purpose to explore the current Business unit involvement in the DevOps activities and mechanisms that are in place to facilitate that.

1. Which activities in a DevOps lifecycle would you identify as the most dependent on the Business unit and the related stakeholders?
    a. Could you provide me with some examples?
2. Could you describe, according to your role, the activities that take place, after a software release?
    a. What is measured?
    b. Are there some KPIs that you monitor?
    c. Who is involved?
    d. Do you identify any factors that hinder this process?
3. According to your role, how is the Planning for the next release performed?
    a. How are the requirements prioritized?
    b. Is there a feedback cycle across organizational levels in place?
    c. Do you think there are some challenges in this process?
4. How is the Customer involvement ensured throughout the agile development lifecycle?
    a. Can you provide some examples?

B.1.4 BizDevOps

To wrap up, a few questions will be asked about the BizDevOps concept, with the purpose to get an overview of how it is currently understood in practice.

1. Have you heard of the term BizDevOps?
    a. What does it mean to you?
2. Have you heard of the Scaled Agile Framework (SAFe)?
    a. If yes, do you find any similarities with the BizDevOps concept?

Do you have any further comments or questions? Please, feel free to contact me anytime for any questions regarding this research. I would like to thank you for providing me with this valuable information!

*** Stop Recording ***

Would you like to suggest somebody else that might be interested in this research? Would you be interested to hear back about this research? Later, I will host a round table with a group of experts to discuss my results. If you are interested, I will send you an invite. Thank you very much for participating!

Appendix C

Treatment Design: Interview Protocol

An Informed Consent, same as in Appendix B has been used when performing the second round of interviews.

C.1 Interview Protocol

The interview will take around 30 to 45 minutes. I would like to record this interview, for the purpose of further analysis. All the information will remain confidential. May I start the recording?

*** Start Recording ***

I would like to inform you that from now, our conversation is recorded.

C.1.1 Introduction

1. Would you please introduce yourself?
   a. What is your role and experience?
2. What does DevOps mean to you?
   a. Why would organizations adopt it?
   b. What benefits of adopting DevOps do you value most?
3. What are 5 things that come to your mind when talking about DevOps?
4. Can you describe the typical roles in a DevOps team?
5. From your experience, do you think there is a gap between the Business unit and software delivery cycle, in companies adopting DevOps?

C.1.2 BizDevOps and the role of Requirements Engineering

6. Have you heard of the term BizDevOps?
   a.What does it mean to you?
7. How is the Business department communicating with the DevOps team?
   a. Can you think of some mechanisms that enable that?
   b. What roles facilitate this communication?
   c. How is the role of the Product Owner involved?
   d. How is the role of the Business Analyst involved?

8. How are the requirements for the next release captured?
   a. Can you name some activities that are performed?
   b. Who is involved in this process?
   c. What tools are being used?

9. What is the origin of these requirements?
   a. How is the customer perspective captured?

10. How are the requirements prioritized?
    a. Who is involved?

C.1.3 Feedback loop in BizDevOps

11. Can you describe what feedback loop(s) is(are) in place?
    a. Who is part of this feedback loop?
    b. How often does it occur?
    c. What are some KPIs that organizations are interested to measure?

12. If you think of the DevOps lifecycle, what processes do you think should be in place, to remove the challenges originating from Business - DevOps silo?
    a. What do you think can be improved or added to the DevOps practices?
    b. What could be the benefits of a successful alignment?

Do you have any further comments or questions? Please, feel free to contact me anytime for any questions regarding this research. I would like to thank you for providing me with this valuable information!

*** Stop Recording ***

Would you like to suggest somebody else that might be interested in this research? Would you be interested to hear back about this research? Later, I will host a round table with a group of experts to discuss my results. If you are interested, I will send you an invite. Thank you very much for participating!

Appendix D

Interviews and Focus group participants references coding

The following tables report on documentation of the interviews and focus group session. All the subjects that participated on this research are assigned to a reference code, which is used for easier citation, over this thesis report. The APA reference is also provided. The name of the participants has been replaced with his/her title, to ensure confidentiality. The column *Round* indicates which protocol was used for the interview. Round I uses the protocol as in Appendix B and Round II uses the protocol as in Appendix C. Table D.2 provides the coding for the focus group participants. Although the session attended 7 participants, only 5 of them were experts in the field, thus reference coding is assigned only to them.

TABLE D.1: Coding scheme used for Interviews conducted for collecting evidence for this research

Code	APA reference	Round
iv1	(Researcher, personal communication, February 19, 2019)	I
iv2	(Agile Coach, personal communication, February 25, 2019)	I
iv3	(Delivery Manager, personal communication, March 6, 2019)	I
iv4	(DevOps Lead, personal communication, March 8, 2019)	I
iv5	(DevOps Lead, personal communication, March 29, 2019)	II
iv6	(Scrum Master, personal communication, March 29, 2019)	II
iv7	(Operations team member, personal communication, April 5, 2019)	II
iv8	(Agile Coach, personal communication, April 10, 2019)	II
iv9	(Performance Engineer, personal communication, April 17, 2019)	II
iv10	(Innovation & Software Architect, personal communication, April 26, 2019)	II

TABLE D.2: Coding scheme used for the participants of the Focus Group validation session

Code	APA reference
participant1	(Developer, personal communication, June 7, 2019)
participant2	(Data Science Consultant, personal communication, June 7, 2019)
participant3	(Development Analyst, personal communication, June 7, 2019)
participant4	(Development Consultant, personal communication, June 7, 2019)
participant5	(Development Analyst, personal communication, June 7, 2019)