UTRECHT UNIVERSITY

MASTER THESIS

DevOps implementation framework for Agile-based large financial organizations

Author:
Anitha Devi Nagarajan
5588790

Supervisors:
Dr. Sietse J. Overbeek
Dr. Fabiano Dalpiaz

A Master thesis document submitted regarding the requirements for the degree of Master of Science

in the

Business Informatics
Department of Information and Computing Science

July 10, 2018

UTRECHT UNIVERSITY

Abstract

Business Informatics

Department of Information and Computing Science

Master of Science

DevOps implementation framework for Agile-based large financial organizations

by Anitha Devi Nagarajan

Modern large-scale financial organizations show interest in embracing DevOps way of working in addition to Agile adoption. Implementing DevOps over agile helps to enhance certain agile practices while extending others. Although there are quite some DevOps maturity models available in the literature, they are neither specific to large-scale financial organizations nor they include the Agile aspects within their scope. Thus, the current study has been performed to identify why such organizations are interested in implementing DevOps and how their implementation can be guided by a conceptual framework. As a result, this thesis presents a list of drivers, a generic DevOps implementation framework and its variations per driver. The development of these artifacts has been guided by design science methodology and they have been validated by practitioners from financial organizations in the Netherlands. The practitioners have identified the developed artifacts as useful to educate their people within their organizations. Moreover, the artifacts have been applied to a real organization's goals to demonstrate how they can be of help to identify the useful measurement units, which in turn can help to measure and achieve their DevOps transformation goals. Thus, the developed artifacts are not only serving as a baseline for the future research but are also useful for the current organizations to commence and get ahead with their DevOps implementations.

Keywords: DevOps implementation framework, Large-scale financial organizations, Agile, Drivers, Measurement units

Acknowledgements

This report is an outcome of my master thesis project work done in the period of 2017 - 2018, as part of my study in Business Informatics program at Utrecht University. I am very satisfied with the progress and outcome of this study, which is not possible without the cooperation and support of several people and their organizations. It is my pleasure to take this as an opportunity to thank them for their assistance and contribution towards this study.

I owe my deepest gratitude to my first supervisor Dr. Sietse J. Overbeek, who has supported me in many number of ways. Without his encouragement, confidence on my work and timely and constructive feedback, this study would have been hardly completed.

I am also grateful to my second supervisor Dr. Fabiano Dalpiaz for offering his help on my thesis whenever I asked.

It is an honor for me to acknowledge the various aids that I have received from Mr. Roy van Dijk, and I thank him for his suggestions and feedback throughout the planning and development of this thesis work. His willingness to give his time so generously has been very much appreciated.

A very special gratitude goes to my friend, Lamia Soussi for sharing the motivation and enthusiasm during the whole period of our theses. I will definitely miss the time that we spent working together. Many thanks for working with me!

Last but foremost, I can not thank enough my husband, Manivannan Gandhiraj and my little daughter Sahana for their invaluable love, encouragement and unparalleled support in all possible ways throughout the period of my thesis works.

Contents

List of Figures

List of Tables

Management Summary

The large financial organizations are originally hierarchical, multi-cultural and their technical dependency is very diversified. Nevertheless, due to the proven advantages with Agile and DevOps adoption, they are willing to embrace these new ways of working within their organizations. The current study is performed in order to help them to become more aware about DevOps adoption and to guide their goal driven DevOps implementations in a theoretical manner.

For that, this research project has developed certain artifacts namely, a list of drivers, a generic DevOps implementation framework and its variations per driver. These artifacts are gained during our path towards answering the main research question, *"how to design a DevOps implementation framework that is helpful for Agile based large financial organizations"*.

The design science methodology has been chosen as the guiding framework to conduct this study and so the research has gone through 4 iterative phases namely, problem investigation, solution design, solution validation and solution implementation. During these phases, the scientific literature have been approached to gather knowledge from the past studies, the semi-structured interviews are conducted to gather information from practitioners and the directed content analysis approach has been followed to develop the mentioned artifacts.

After the development of the artifacts, they have been evaluated by 5 practitioners by means of 3 validation sessions. They have been presented with the artifacts and asked to assess them against several criteria such as completeness, usefulness and fit with the organization. In between the sessions, the possible comments are incorporated in the artifacts so that the updated artifact is used for the following validation session. Most of the practitioners agreed that the developed artifacts provide a complete and high-level overview of the DevOps implementation, which is useful for the organizations, where they belong to. They also agreed that this framework is beneficial to upskill the people within their organizations regarding the components within the DevOps implementation and their significance.

Additionally, the developed artifacts have been applied to a real organization's DevOps implementation goals to derive the useful measurement units that are helpful to measure their progress towards their goals. This scenario has been demonstrated to justify the usefulness of the developed artifacts in a real world scenario. However, this demonstration has not been implemented in the actual organization due to the time restrictions and so the true implications have not been gathered or measured.

1 Introduction

1.1 Background

The Agile software development methodology, which is based on a defined set of principles and values, encourages more self-driven, collaborative, customer-oriented and incremental software development (Beck et al., 2001). In the past decade, Agile methods have gained a widespread acceptance among all types of organizations that involve software development. This is because of proven advantages like faster software development with improved quality, and the ability to welcome changes throughout the project leading to improved customer satisfaction in comparison with traditional software development approaches such as the waterfall method (Bhadoriya et al. 2014) and the incremental method (Tarhan and Yilmaz 2014). Nevertheless, the structural division between the software departments such as development and system operations remained the same with their different, yet conflicting needs and priorities (Iden et al. 2011; França et al. 2016). This division leads to delays in the deployment of the developed software in the actual production environment, which results in longer release cycles (Ravichandran et al. 2016a). As a solution addressing this issue, the DevOps movement has emerged with the purpose of closing the gap between development and operations and it has evolved mainly from cloud based product organizations (Wettinger et al. 2014).

DevOps practices have evolved through cloud-based product organizations like Facebook, Flickr and Netflix. The similarity among these examples is that they are having one big complex online software product, which is not the case with financial organizations such as banks that have multiple interconnected systems. Moreover, their software mainly concerns core banking applications that are built on old technologies such as mainframes, but they also utilize new technologies for applications like mobile banking applications. In this scenario, bringing changes into the depending technology or underlying system architecture in order to gear up for DevOps is not desired by such financial organizations since they prominently involve cloud (Zhu et al. 2016). Besides the internal issues, bankers consider data security, confidentiality and regulatory restrictions, the major barriers for cloud adoption (Hon and Millard 2016).

Yet, the DevOps paradigm is not exclusive to organizations that are surrounded with cloud computing (Wettinger et al. 2014). Ravichandran et al. (2016) have given an example success story of a large Dutch bank, which successfully embraced DevOps and achieved potential benefits in support of their DevOps efforts such as continuous delivery and automation. By bringing in 'Scrum' teams that involve both development and operations competence and by employing automation to speed up the possible activities including Configuration Management and Software Deployment they increased the application release frequency from an average of once in every 13 weeks to weekly updates. In case of mobile applications, they accelerated the code

deployment status from several hundreds to 80.000 deployments per month. This exemplifies how large financial organizations can achieve big success by successfully implementing DevOps.

The objectives behind a DevOps implementation may differ considerably among different types of organizations and so the corresponding means to measure their success. Hence, it is more important to identify the goals and objectives of implementing DevOps, which in turn will help to find out a suitable approach to meet their goals and appropriate metrics that can measure their success against the intended objectives (Elberzhager et al. 2017; Erich et al. 2014b). This variability among the needs of different organizations leads to various definitions for DevOps and approaches towards a DevOps implementation (Geurts 2016; Erich et al. 2014a). Yet, the usage of a DevOps implementation framework can accommodate this uniqueness by allowing organizations to take their own approach towards a successful DevOps implementation.

1.2 Problem Statement

DevOps is often coalesced with Agile principles and practices. Those studies that compared DevOps with other software development methodologies have identified that both Agile and DevOps have similar goals and values, but their scope varies. For instance, both of them encourage transparent and open communication. DevOps does this mainly between Development and Operations but is not limited with them; whereas, Agile does that among the team and with other stakeholders. It has been described that DevOps is an extension of Agile software development since Agile adoption is a key factor for successful DevOps adoption (Lwakatare et al. 2016b). When DevOps is laid over an Agile implementation, it enhances several Agile practices while extending others outside development activities (Huttermann 2012; Lwakatare et al. 2016a; Jabbari et al. 2016). There are several studies that list out the factors required for a successful agile transformation from various perspectives (Darwish and Rizk 2015; Hameed et al. 2016; Chow and Cao 2008). The existing DevOps implementation models have focused only on DevOps and so little attention has been paid on its relationship with Agile principles and practices (Bucena and Kirikova 2017; Mohamed 2016). This emphasizes the need to focus on both Agile and DevOps in order to bring up a solution that unifies the similarities, replacing the outdated Agile practices with enhancements endorsed by DevOps. Moreover, to the best of our knowledge, there has not been found any specific study focusing on DevOps implementation practices of financial organizations. The needs and expectations of every organization is different regarding the DevOps implementation within their organization (Erich et al. 2014b; Elberzhager et al. 2017) and so is their goals and metrics. The existing studies have concluded with the development of conceptual frameworks whereas it is quite important for the organizations to know how these frameworks can be of help to achieve their goals.

Motivated by these concerns and encouraged by the current needs of participating organizations, it is intended to develop a conceptual framework, which depicts the various aspects involved in the DevOps implementation of Agile organizations in terms of perspectives and focus areas, but the framework needs to be specific for

financial organizations and their drivers need to be understood. Moreover, every transformation is backed up with certain goals and their progress is measured with certain metrics. Thus, the intended framework should be developed to help organizations with their goal-oriented DevOps implementations.

1.3 Research Objectives

The study has two important objectives to accomplish as mentioned below:

Objective 1: Develop a DevOps implementation framework suitable for Agile software development units within financial organizations.

Objective 2: Reveal the relationships among the main drivers, the developed DevOps implementation framework and the metrics.

1.3.1 Research Questions

In order to systematically attain our objectives, we have formulated a main research question (MRQ) and several sub research questions (SQ) as detailed below.

Main Research Question: "How to design a DevOps implementation framework that is helpful for Agile based large financial organizations?"

SQ1: Why are financial organizations interested in implementing DevOps besides following Agile principles within their software development units?

In the first phase of this thesis, we try to identify the drivers that motivate financial organizations to implement DevOps along with Agile. By reviewing the related scientific literature (Chapter 2) we will understand the overall motives and goals related to DevOps adoption in general. Nevertheless we also perform the semi-structured interviews (see Appendix A for the interview protocol) with people working at banks in order to their specific goals so that the main drivers for the model can be identified.

SQ2: What are the different perspectives and the main focus areas that are required for a DevOps implementation while following Agile principles at financial organizations?

Along with the drivers mentioned in SQ1, we also intend to identify the different dimensions of the DevOps transformational factors and the main factors themselves within those dimensions. Such dimensions will be classified as 'perspectives' and

the main factors within those perspectives will be called as 'focus areas'. As mentioned in SQ1, the literature review will be utilized to understand the existing classification of perspectives related to the DevOps transformation and the main transformational factors within them. Later, the interviews conducted with the practitioners will be used to gather the industry-specific information which will be used later as a base for the development of the competence model.

SQ3: How does a conceptual framework incorporating identified perspectives and focus areas look like?

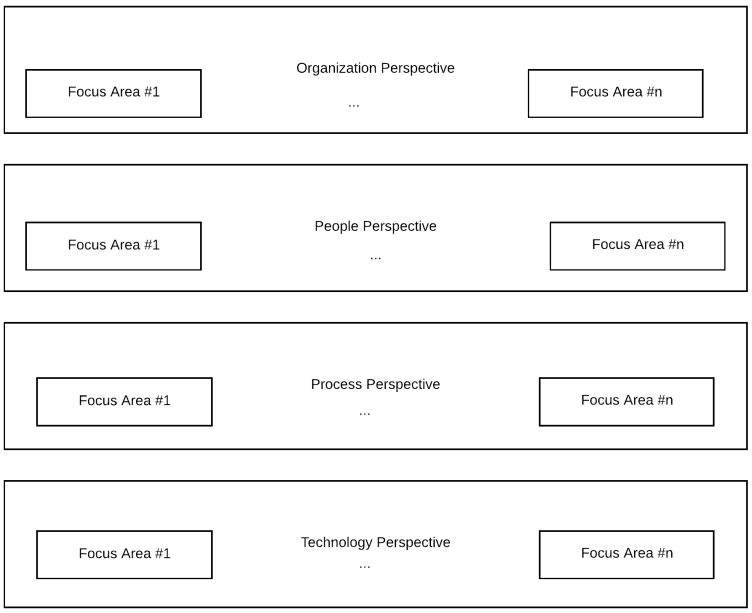

FIGURE 1.1: A sample template for a prospective DevOps implementation framework

As Figure 1.1 depicts, those perspectives and focus areas that are identified from both literature reviews and practitioner interviews will be utilized to develop the DevOps implementation framework, which outlines various perspectives and their corresponding focus areas specific to the Agile based DevOps implementations at financial organizations.

In the given Figure 1.1, people, process, product and organization are the conceptual perspectives. Each of the perspectives has several focus areas, which are the important themes that need to be present within those perspectives. For example, organization perspective can be identified with focus areas such as organizational culture and organizational structure.

SQ4: How can the relationship between identified drivers and the developed DevOps implementation framework be revealed?

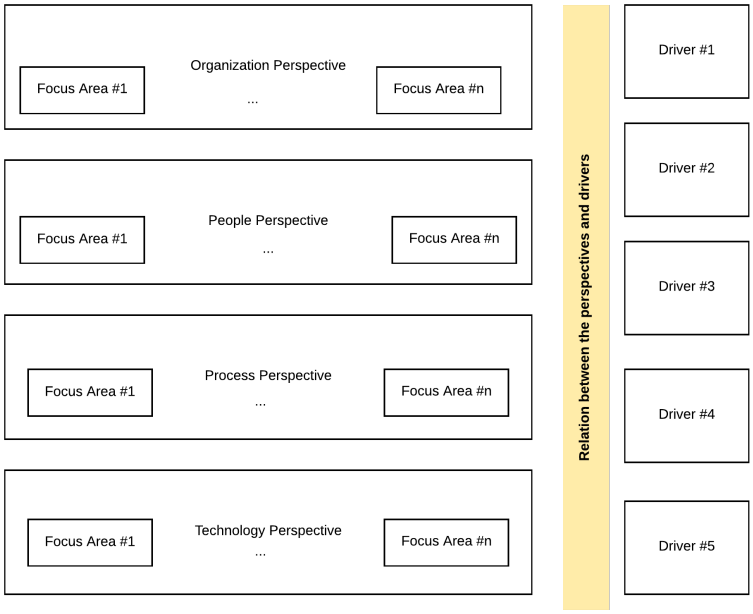

FIGURE 1.2: A visual representation of the prospective DevOps implementation framework and its relationship with the drivers

In this question, we will try to identify the relationship between the identified list of and drivers and the competence model so that the organizations can make the best use of the model to achieve their goals, which in turn are related to the drivers. This will help the organizations to identify which factors are potential for every driver. The representation given in Figure.1.2 illustrates the purpose of this sub-question. This representation is inspired from the DevOps implementation framework developed by Feijter et al. (Feijter et al. 2018).

SQ5: How can the identified drivers, the developed DevOps implementation framework, and their relationship help to identify the relevant measurement units?

The last question discusses on identifying the right measurement units which will be helpful to measure the transformational efforts and success based on the developed framework and the corresponding drivers. In order to answer to this question, a case study will be performed at one of the participating organizations so that the developed artifacts will be applied at their organizational goals to identify the possible useful measurement units.

1.4 Thesis organization

This thesis report documents every process aspect of the undertaken work and outcome of those processes. The structure of the report is given below in order to guide the reader to navigate through the document.

Chapter 2: Research Design. This chapter elaborates how the design science framework has been utilized to the execution of this study while describing the exercised research methods and techniques. In addition, this chapter includes a Process Deliverable Diagram (PDD) of this study to show the main processes and their resulting deliverables and their relation to this report.

Chapter 3: Results of the literature review. The relevant scientific literature regarding Agile and DevOps are reviewed in this chapter to provide a solid baseline to the current study.

Chapter 4: Current Agile and DevOps practices at Dutch Banks. The summary of the data collected via practitioner interviews are summarized in this chapter to show the state of the practices.

Chapter 5: Towards the development of the artifacts. In this chapter, the relevant components of the artifacts such as drivers, perspectives, focus areas and their relationships are identified and documented. Later in that chapter, the artifacts are constructed by positioning the relevant components.

Chapter 6: Application of the artifacts to a real world scenario. In order to rationalize the usage of developed artifacts, they have been applied to a real organization scenario and explained how they can be used to derive the useful measurement units based on the organizational goals.

Chapter 7: Artifacts validation. The evaluation scenario in terms of evaluators' profile and evaluation criteria, and the evaluation results are described in this chapter.

Chapter 8: Results and Discussion. This chapter summarizes the results of this study by answering all the research questions presented in Chapter 1 and presents the discussion of those results as well as the limitations of the current study.

Chapter 9: Conclusion. This last chapter concludes the report by mentioning the main contributions of this study and details the future research indications.

2 Research Design

This section describes the research design by detailing the research framework and the involved research methods and techniques. The design science method (Wieringa 2010) is considered for the high level design of this study and so the relevant phases of the project are explained accordingly. The research methods utilized for data collection, data analysis and artifact validation are explained by drawing the association to the sub research questions introduced in Chapter 1. Additionally, the research plan is drawn to indicate the prospective project milestones and the corresponding time line.

2.1 Research Framework: Design Science Methodology

As mentioned earlier in Chapter 1, the current study aims to achieve the following research objectives.
Objective 1: Develop a DevOps implementation framework suitable for Agile software development units within financial organizations.

Objective 2: Reveal the relationships among the main drivers, the developed DevOps implementation framework and the metrics.

In order to achieve these objectives, the design science methodology (Wieringa 2010) has been adopted. The adaptation of the engineering cycle from design science approach followed for this study is shown in 2.1. As can be seen in the Figure 2.1, the complete engineering cycle is not utilized in this study since the implemented solution is not evaluated after application and so the cycle is not continued based on the user experience after implementation due to the time restrictions. However, the design cycle, a smaller part of the engineering cycle that iterates among problem investigation, solution design and solution validation phases is still utilized for the development of the research artifacts that are listed in Table 2.1.

Problem Investigation:

The first phase of problem investigation is used to get an outright understanding of the phenomenon under study (Wieringa 2010). First, the research objectives and research questions are formulated based on the interests of the involving stakeholders. Next to that, the literature study is carried out to review other relevant scientific studies to lay a stable theoretical foundation. Finally, semi-structured interviews are conducted with practitioners from different banks in order to collect industry specific data. Furthermore, the requirements about the prospective artifacts are also collected from the stakeholders.

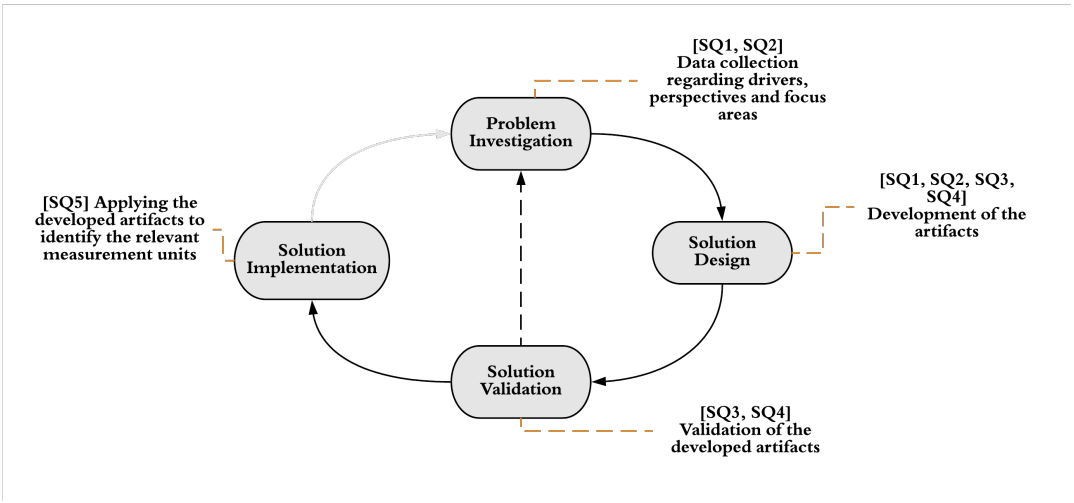

FIGURE 2.1: The engineering cycle adaption (Wieringa 2010)

Solution Design:

The second phase is concerned with the design and development of relevant artifacts (mentioned as treatment in (Wieringa 2010)) that could treat the problem characterized in phase 1 (Wieringa 2010). The main artifact that this current study aims to develop is a DevOps implementation framework, an abstract model with its documentation describing the main perspectives and the corresponding focus areas to help large agile based financial organizations with their DevOps implementations. For the other related artifacts, please refer the table. 2.1.
The directed content analysis approach is adopted to explore the data gathered from literature and practitioner interviews in the first iteration and added expert interviews in the later iterations. This approach allows utilizing the relevant categories such as drivers and focus areas within the main artifact as the structure is adopted from (Feijter et al. 2018).

Solution Validation:

The third phase deals with validating the solution/treatment to verify whether it would satisfy the stakeholder goals if implemented (Wieringa 2010). In the current study, the developed DevOps implementation framework is evaluated by a panel of experts to verify whether it has sufficiently addressed the relevant concepts of Agile and DevOps and to assert that the artifacts are developed enough to be used as intended at the financial organizations. Meanwhile, the other artifacts such as drivers, perspectives, focus areas and their relationships with each other will also be reviewed by the experts during their validation.

Solution Implementation:

The final phase of the engineering cycle allows implementation of the artifact(s) in the problem context (Wieringa 2010). In this study, the designed DevOps implementation framework and their variations per drivers will be utilized to explain how

they can be used to identify the useful measurement units according to their implementation goals. This will be performed by utilizing an explanatory case study, in which one of the participating organization's goals will be utilized to explain the measurement units derivation with the help of developed artifacts.

2.2 Research Methods and Techniques

The research framework section was explained with several methods and techniques to be used in this research. This section provides a detailed explanation of the mentioned methods and their rationality towards this research. To provide a clear understanding of the various methods used in this research and their logical connections with the research questions, Table 2.1 has been displayed with the relevant details.

Sub-Research Question	**Data Collection Method**	**Data Analysis Method**	**Resulting Deliverables**	**Deliverable Evaluation Method**
[SQ1] Why are financial organizations interested in implementing DevOps besides following Agile principles within their software development units?	Literature Review Interview	Content Analysis (Nvivo)	Drivers	Expert Opinion
[SQ2] What are the different perspectives and the main focus areas that are required for a DevOps implementation while following Agile principles at financial organizations?	Literature Review Interview	Content Analysis (Nvivo)	Perspectives & Focus Areas	Expert Opinion
[SQ3] How does a conceptual framework incorporating identified perspectives and focus areas look like?	-	-	DevOps implementation framework	Expert Opinion
[SQ4] How can the relationship between the identified drivers and the developed DevOps implementation framework be revealed?	-	-	DevOps implementation framework variations per driver	Expert Opinion
[SQ5] How can the identified drivers, the developed DevOps implementation framework, and their derived relationship help to identify the relevant measurement units?	Explanatory case study	-	Case study report	Peer Review

TABLE 2.1: Research questions and methods association matrix

2.2.1 Literature Review

There are substantial number of studies available on Agile in comparison with the available number of scientific literature focusing on DevOps. Both the subjects are included for the literature research to understand their key principles, organization's drivers to adopt them, factors involved in their implementation within an organization, and the existing maturity models focusing on Agile and/or DevOps. There were several search engines utilized for gathering the scientific knowledge base namely, Computer Science Bibiliography (dblp.org), Google Scholar, ScienceDirect and Scopus. The articles were selected based on their scope to make sure that they include atleast one of the mentioned subtopics within Agile or DevOps. Then, the selected articles are entered into the concept matrix excel sheet based on their inclusion of topics as explained later in this section.

In order to identify the Agile related articles from the mentioned search engines, the following keywords were used to select given types of articles.

Keywords: {Agile software development} + {principles or drivers or objectives or goals or benefits}, {implementation factors or success factors}, {maturity or maturity model or maturity framework}, {DevOps}

Types : Conference proceedings, books, journals, thesis documents

Likewise, the following details were used in case of DevOps.

Keywords: {DevOps} + {definition or explanation}, {drivers or objectives or principles}, {implementation factors or practices or success factors or culture }, {maturity or maturity model or maturity framework}, {Agile}

Types: Conference proceedings, books, journals, thesis documents

In addition to the papers identified by above search terms, several other papers were identified by backward snowballing approach, which helped to identify relevant articles from the references of other articles found (Jalali and Wohlin 2012). This helped us to accumulate a solid scientific knowledge base required for this study.

The articles identified as above described were constantly compared and organized into an excel sheet to build the concept matrix as illustrated in the Figure 2.2. The literature articles were mapped out to the related concepts such as Agile drivers, Agile advantages, Agile adoption factors, DevOps drivers, DevOps adoption factors etc. This helped us to review the literature throughout the study to make sure that the relevant details have been obtained during the various phases involved.

Article #	Concept 1	Concept 2	...	Concept m
Article 1	✓			✓
Article 2		✓		✓
...				
Article n	✓	✓		

FIGURE 2.2: The illustration of concept matrix adapted from (Webster and Watson 2002, p.17)

2.2.2 Interviews

The second data generation method used is interviews. Oates (2005) defines an interview as a conversation of specific kind which has a set of unspoken assumptions, involves an agenda in general and is led by the researcher. Moreover, he acknowledged three types of interviews based on the format. First is the structured interviews in which the interviewer will be asking a predetermined set of questions to every interviewee without involving in any conversation. Second is the semi-structured interviews, in which the interviewer will still be having a set of questions but always involve in a conversation with the interviewee and thus may change the order and/or content of questions based on the flow. The third one is the unstructured interviews which gives more control to the interviewees to talk unrestrained

about the topic of interest. Based on the number of interviewees, interviews can be further classified as individual or group interviews (Oates 2005).
In this study, semi-structured individual interviews will be adopted as an approach to collect data from the practitioners. Due to the flexibility to conduct the interview according to the flow of conversation with the interviewee, this approach is found to be the best suitable option for this study, since it allows the researcher to have more control on leading the conversation based on the pre-acquired knowledge from literature and other interviews. Please refer to Appendix A and Appendix C for the interview protocols used for problem investigation and expert opinion respectively.

2.2.3 Content Analysis

Content analysis is a systematic approach of bringing out meaningful interpretations from the text data and so is utilized as such to analyze the data collected from both literature reviews and interviews. There are three different content analysis approaches observed namely, conventional, directed and summative. Conventional content analysis allows categories and their names to emerge from the content of the data being analyzed. On the other hand, directed content analysis is a more structured process that utilizes key concepts or variables from existing research. At last, summative content analysis identifies and quantifies certain content in the text under analysis in order to understand its contextual use (Hsieh and Shannon 2005).

As mentioned in section 2.2.4, this study utilizes the general structure and categories from (Feijter et al. 2017) such as drivers, perspectives, and focus areas but differentiates in terms of the field of study while extending the scope to include Agile. In addition to that, a list of perspectives are identified based on the literature research performed on the subject of Agile adoption factors. Therefore, further data collection (through interviews) and data analysis are guided to follow the structure identified from the mentioned earlier studies. Due to these characteristics, the directed content analysis approach is found to be useful for the data analysis part of this study. In addition, as indicated in Table 2.1, the data analysis is supported by Nvivo tool.

2.2.4 Expert Opinion

As shown in Table 2.1, all the main artifacts resulting from this study will be reviewed by an expert group, by following the expert opinion method. In this method, the experts will be given with the artifacts, and they are asked to express the effects that they think will happen when the artifacts are used in a realistic environment. When such predicted effects are not compliant with the requirements, the artifact has to be redesigned (Wieringa 2010). Since expert opinion offers an effective way to validate conceptual designs, it is decided to use this method for all the major design based artifacts as it is given in table. 2.1.

2.2.5 Case Study

Case study is typically preferred to answer 'why' or 'how' type of questions. Case study can be either of these : explanatory, case study helps to disclose a phenomenon; exploratory, case study that delves into the reality; or, descriptive, case study that

articulates 'what is' already known. Case studies can utilize either or both quantitative and qualitative methods (Yin 2011). The current study considers explanatory case study to answer sub-research question SQ5 as shown in Table 2.1. In this case, the case study can utilize the organizational goals of one of the participating organizations to explain how the developed artifacts can be utilized to identify the relevant measurement units, which in turn help to measure their DevOps adoption progress towards their goals.

2.3 Research Execution

This section describes how the research is executed in terms of involved processes and the deliverables corresponding to those processes. In order to concisely show this, a process deliverable diagram (PDD) has been developed as shown in figure.2.3. This PDD has been developed based on the meta-modeling technique proposed by Weerd and Brinkkemper (2008). PDD is a combination of an activity diagram which shows the processes on the left-hand-side and a deliverable diagram that exhibits the artifacts on the right-hand-side of it.

This PDD consists of the engineering cycle phases namely problem investigation, solution design, solution validation, solution implementation as the main activities besides the research initialization and research finalization activities. The deliverables that result from each of the sub-activities within the main activities and their relationships with each other and to the main deliverable, this thesis report are also unveiled in the given PDD.

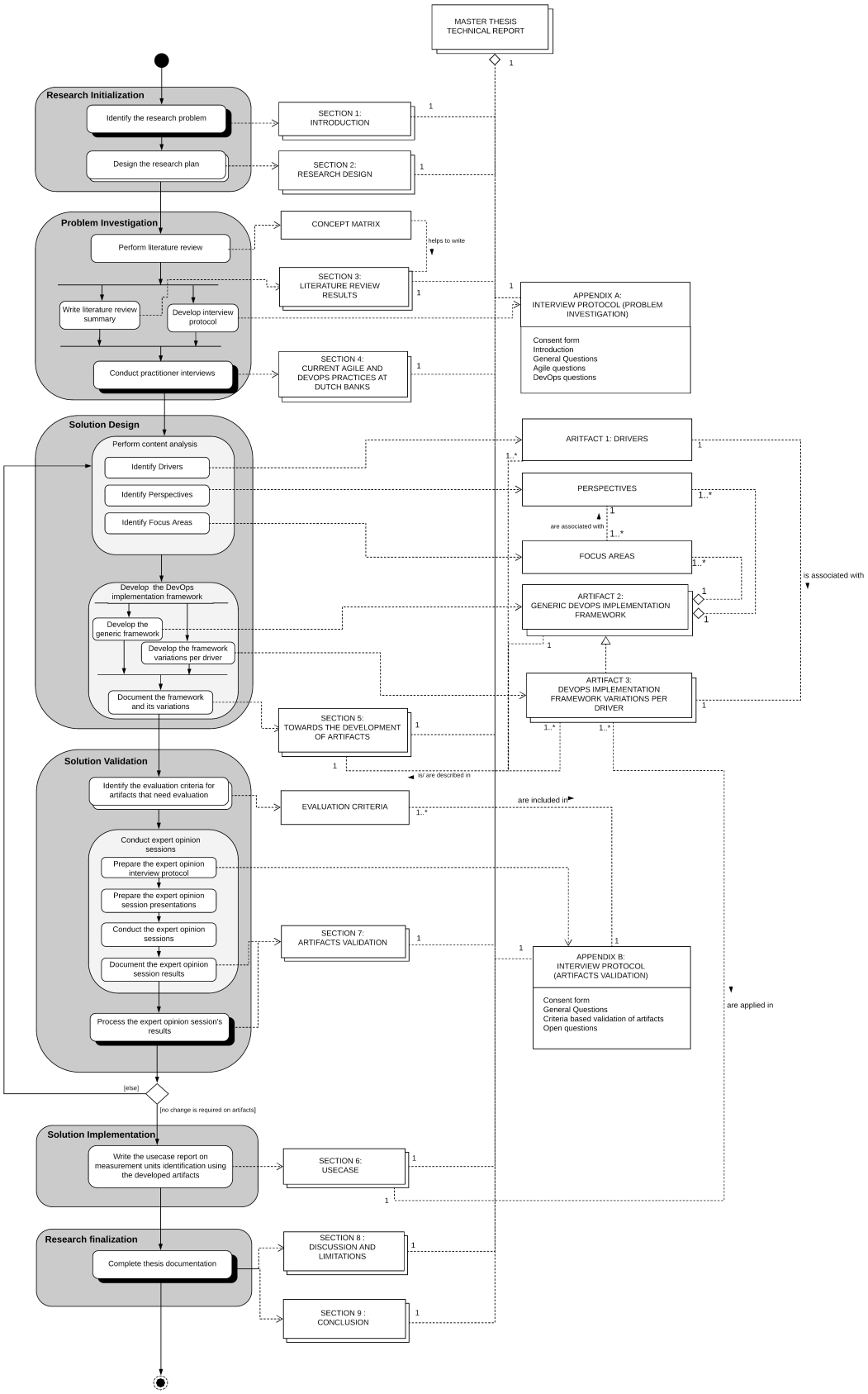

FIGURE 2.3: A process deliverable diagram of the main processes and deliverables of the current study

3 Results of the literature review

In this section, existing scientific literature from related topics are reviewed in order to gain a solid background and deeper knowledge on the research topic and to understand what has already been done so that the existing knowledge can be utilized well for this study. Therefore, we considered to review the topics - Agile and DevOps in order to understand the drivers and factors associated with their implementation. In addition, the existing maturity models and the performance metrics related to Agile and DevOps are also examined.

The main topics (i.e. Agile and DevOps) and subtopics (i.e. Drivers, Factors inclusive of their categories, and Performance metrics) involved in the literature study along with their identified relationships with other relevant entities (i.e. organizations, company, and benefits) are illustrated in the following Figure 3.1.

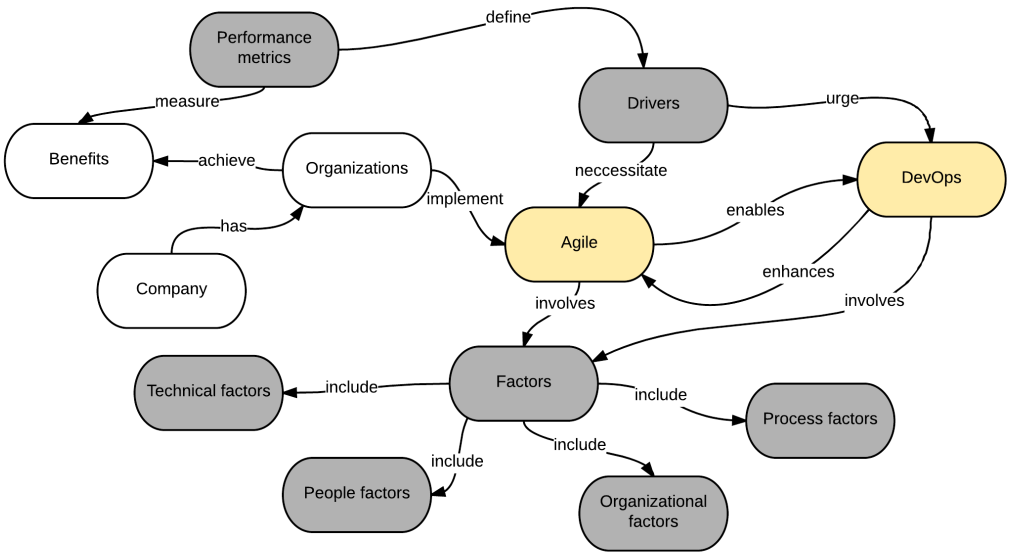

FIGURE 3.1: Main concepts and their relationships resulted from literature review

3.1 Agile

3.1.1 Drivers for Agile adoption

About a couple of decades ago, when the practitioners in the software development industry have started realizing the several detriments of their beloved waterfall method for software development, Agile manifesto has been written by seventeen independent-minded software practitioners. The main disadvantages identified with traditional methods are: (a) the heavy planning required by the projects, which required the detailed and clear requirements at the beginning of the projects; (b) the inflexibility of software development methods which could not accept the changes during the course of the development but before delivery; (c) long development time required by projects to deliver the software which did not satisfy the customer most of the times; and (d) delayed testing of the software, which increased the cost to fix the identified defects. Due to the significance of such drawbacks, the advantages of the waterfall method, such as easy implementation and simple cost estimation, were not appreciated (Adel and Abdullah 2015).

There are a few traditional software development methods which allow iterative fashioned development process like Agile. Tarhan and Yilmaz (2014) summarized the characteristics of well-known traditional software development methods (i.e. Waterfall, Iterative, Evolutionary and Spiral) and Agile methods, namely XP (Extreme Programming), FDD (Feature Driven Development), Crystal, Scrum, DSD (Dynamic System Development), Lean Software Development and UDP (Unified Development Process). This summary helps differentiating the Agile methods from iterative traditional methods. Moreover, the authors performed a comparative case study on traditional incremental software development process and Agile process at the same company to identify the differences achieved in the end result. According to the results of that study, Agile processes were found to produce better quality software in less time in comparison with the traditional incremental method. Such empirical studies justify the reason for organizations to transform their software development method to Agile.

Tseng and Lin (2011) claim that the current business environment faces various changes every day and so they are supposed to have the ability to respond to those changes as early as possible. This necessitates them to become Agile enterprises. Moreover, the authors have summarized the general areas in which change is encountered by such enterprises as following: (a) Technological innovation; (b) Volatile markets; (c) Dynamic customer requirements; (d) Changes in social factors such as workforce expectations and legal pressure; and (e) Competition criteria. Agile enterprises are identified with following capabilities that help them to face such uncertainties: (a) Responsiveness, the ability to identify changes and respond to them (b) Competency, the efficiency and effectiveness of an enterprise in reaching its goals (c) Flexibility to achieve different goals by implementing different processes and (d) Speed.

3.1.2 Factors associated with Agile adoption

In order to achieve success with an Agile adoption, it is required for an organization to ensure that certain aspects are in place. These factors determine the success or

failure of the corresponding Agile implementation. Several authors have explored such factors and many of them classified their findings into dimensions namely: organizational, people, technical and process (Darwish and Rizk 2015; Hameed et al. 2016; Chow and Cao 2008). In order to get a better understanding, these literature identified factors are explained together with those factors, that are deduced from the Agile principles themselves.

Organizational Factors:

When Agile is adopted by an organization, it does not reflect only among the team members' way of working but also within the mindset of all people including management, and even in the culture of the whole organization. Several studies have identified that corporate culture and organizational environment are the crucial factors of organization-wide success of Agile adoption (Hameed et al. 2016). Agile works better in a collaborative organizational environment where success results through teamwork, which is different from a hierarchical environment where command and control are exploited by the management (Geurts 2016; Iivari and Iivari 2011). In such an environment, small sized teams will be working together by sharing tacit knowledge, mostly in an informal way and being led by a supportive management (Serrador and Pinto 2015). Both researchers as well as practitioners confess that management buy-in is as important as team's commitment and customer involvement in following Agile (Tanner and Willingh 2014; Chow and Cao 2008).

One of the specific ways that management can support teams with Agile adoption is team distribution. As mentioned earlier, Agile puts more value on oral, face-to-face communication over more formal, written communication (**Principle 6**). Thus the teams should be structured in such a way that close collaboration is possible, and be provided with appropriate tools that allow effective communication among people. The other responsibility of the management is to help their organization sustain agility by properly rewarding the Agile practices followed within the team, having a trust in their abilities (**Principle 5**) and motivating them to achieve more by appreciating what have already been accomplished (Hameed et al. 2016; Chow and Cao 2008). One of the interesting findings of Vijayasarathy and Turk (2012) is that larger the organization is and the more experience the developers have, the more difficult it is to adopt Agile. When this is probably the case with any (financial) organization, it is the responsibility of the management to choose appropriate implementation strategies, allocate time, money and resources for all necessary activities including training people to adopt the usage of Agile methods.

People Factors:

This classification is mainly about the characteristics of people in the project team, specifically team members. Agile promotes frequent communication between team and business (**Principle 4**). In order to take advantage of it, team should be capable of communicating effectively with management, customers and other involved stakeholders. Thus it is imperative to have good communication skills by the team members (Darwish and Rizk 2015; Hameed et al. 2016). Additionally, Agile believes in collaborative team work performed by self-organizing teams (**Principle 11**), which is not possible without reporting and adapting within the team (Fontana et al. 2014). Agile endorses frequent and short time-scaled deliveries (**Principle 3**) while

it welcomes changes throughout the development process (**Principle 2**). In order to follow these principles, it is vital to have technically competent teams, who have the ability to handle project complexity well while being aware of the dynamic nature of those projects (Chow and Cao 2008). Agile urges on continuous improvement at a constant pace (**Principle 8 and 12**) and for that, team members should be willing to improve themselves while helping others to grow (Fontana et al. 2014; Tanner and Willingh 2014). Fontana et al. (2014) have summarized the discussed characteristics of Agile teams very well in Table 3.1.

Category of characteristics	**Concept**
Knowledge	Keep lessons learned Knowledge of the customer's business Knowledge of the project Knowledge of the technology Trained team
Behaviour	Collaboration Commitment Making an effort to keep practices in use Self-organization Understand customers
Communication	Communication within the team Communication with customers
Experience	Expertise in Agile practices Time spent working with Agile

TABLE 3.1: Characteristics and related concepts of Team as adapted from (Fontana et al. 2014, p.150)

Next to management and team members, the other important group of stakeholders, who is closely related to the success of any Agile project is customers. Several studies have mentioned that collaborative customer representatives, who have well-defined roles and good involvement in the Agile projects are positively influencing the success of those projects (Darwish and Rizk 2015; Chow and Cao 2008; Tanner and Willingh 2014).

In addition to all these, people should be aware about the uncertain nature of projects and Agile methodology, since lack of Agile knowledge within team, management and customers have been found to negatively influence the success of an organization's Agile implementation (Gandomani et al. 2014).

Process Factors:

Considering the practices that are involved in the 'way of working' as the process dimension, the selection of proper method(s) corresponding to Agile principles is the most important factor identified by several authors such as Chow and Cao (2008) and Hameed et al. (2016). As mentioned earlier, Agile enforces incremental software development (**Principle 8**), whereas the progress is measured with the amount of working software (**Principle 7**). Furthermore, the software development is performed in a consistent manner that helps to learn and improve within every developmental project rather than between one project and another (**Principle 12**). In

order to comply with these principles, it is required to follow effective requirement management process, project reporting and progress tracking mechanisms, project planning process that are all Agile oriented and suitable to integrate with other processes within an organization (Darwish and Rizk 2015; Hameed et al. 2016; Chow and Cao 2008).

In addition to the discussed project management processes, Agile organizations are required to have well-defined project definition processes as well in place. These processes tend to handle activities such as risk management, project time allocation and project status estimation (Darwish and Rizk 2015; Hameed et al. 2016).

Technical Factors:

The technical factors are regarding the usage of suitable Agile practices and tools to complement the various processes involved. Some of such practices are Test Driven Development (TDD), pair programming, code refactoring, and continuous integration. The decision regarding the selection of these practices and related tools should be taken based on the chosen agile method and other process factors, organizational factors and people factors (Chow and Cao 2008; Darwish and Rizk 2015; Hameed et al. 2016). Diebold and Dahlem (2014) performed an exploratory study to identify the state-of-the-art Agile practices by analyzing 68 projects. It identified several practices that are fully or partially being used within the Agile organizations that correspond to various domains such as Finance & Insurance, Consulting, Governance and Telecommunication. Some of such identified practices are time-boxing (which is the most used practice out of the study), learning loop, planning meeting and daily discussion.

Before summarizing the various factors discussed, it could be useful to highlight the Figure 3.2, which shows the important concepts that define the maturity in Agile Software Development. Following that, Figure 3.3 highlights how the concepts from Figure 3.2 associate to the Agile adoption factors discussed above in this subsection.

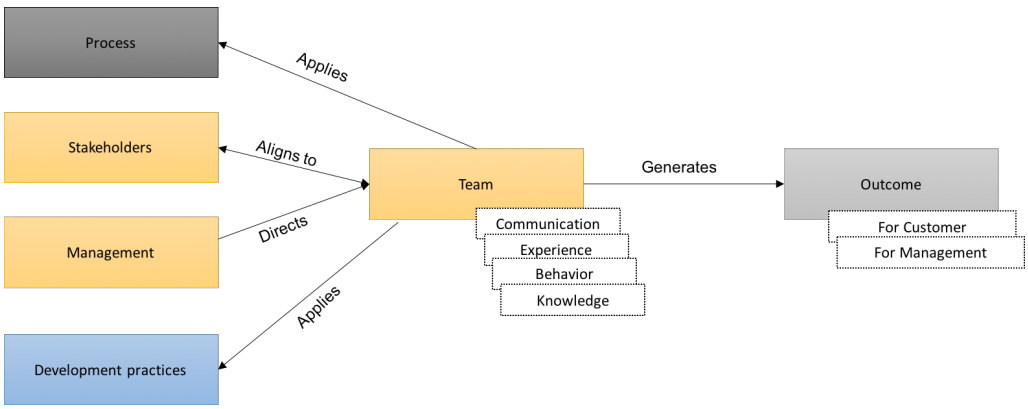

FIGURE 3.2: Relationship among concepts that define Agile Software Development maturity adapted from (Fontana et al. 2014, p.152))

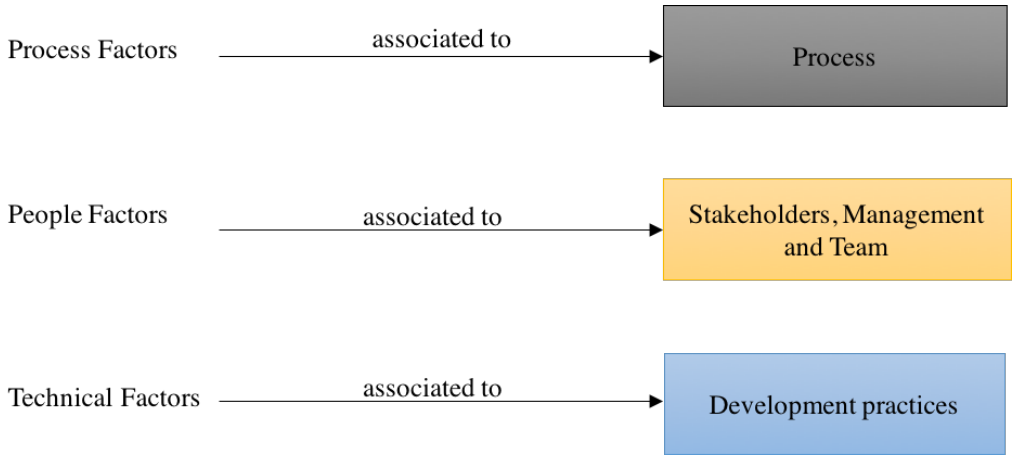

FIGURE 3.3: Association between the discussed factors and the concepts defined by (Fontana et al. 2014))

3.1.3 Agile Maturity Models

The knowledge of existing Agile maturity models is also important for this study since it considers Agile concepts as well into the development of the prospective maturity model. Leppanen (2013) performed a comparative analysis on eight existing Agile maturity models based on several aspects such as purpose, structure and approach. As per that, the compared models are concentrating on the adoption of agile values, principles and practices but on a very high level. The models consist between 4 and 6 levels which are arranged in a way that the lowest level (i.e. level 1) defines the easy to adopt principles and practices whereas, the highest level spans its focus towards the entire organization or of something larger in scope.

3.2 DevOps

3.2.1 What is DevOps?

Several researchers have claimed that DevOps has no clear definition (Lwakatare et al. 2016a; Jabbari et al. 2016; Smeds et al. 2015) and so several interpretations of it are observed among practitioners (Erich et al. 2017). Some of such interpretations are considering DevOps as a job title or role or a separate team within an organization (Lwakatare et al. 2016a; Erich et al. 2017). However, most of these interpretations are actually misinterpretations about DevOps, because it is neither a job title nor a specific team within an organization; it is neither only about tools nor just about startups (Davis and Daniels 2015). This lack of certainty about the term leads to obscurity and often hinders its adoption (Erich et al. 2014b). According to a study by Lwakatare et al. 2016, most of the practitioners refer DevOps as a cultural and professional movement, whereas many of them consider it to be a set of practices, culture, approach, philosophy and mindset.

In current literature, multiple scholars are attempting to bring up suitable definitions for DevOps. Jabbari et al. (2016) performed a systematic literature study to find out the most occurring components within DevOps definitions found in the literature

studies available until 2016. They identified that communication, collaboration and team working, 'bridge the gap', development method, software delivery, automated deployment, continuous integration, and quality assurance are key terms next to development and operations. However, França et al. (2016) do not agree with considering DevOps as a method or methodology, since these terms involve presence of a systematic approach, which is absent in case of a DevOps implementation.

Dyck et al.(2015) have defined DevOps as an organizational approach that motivates cross-functional collaboration between software development and IT operations teams to speed-up the delivery of changes done in resilient systems and to operate them. Lwakatare et al. (2016) have fine-tuned the same definition by clarifying that the DevOps approach constitutes a mindset and a related set of practices.

França et al. (2016, p.56) have generalized the group of individuals participating in the DevOps phenomenon by defining it as a "movement of ICT professionals addressing a different attitude regarding software delivery through the collaboration between software systems development and operation functions, based on a set of principles and practices, such as culture, automation, measurement and sharing". Thus, it can be clarified that DevOps is not restricted only to Developers and Operations team members of an organization but also to other extended team roles such as architects and product owners (Swartout 2014).

3.2.2 Drivers for DevOps Implementation

Iden2013InformationDevelopment(**Iden2013InformationDevelopment**) conducted a Delphi study with experts in system development, IT operations and system owners in order to understand the problems that occur between development and operations personnel in development projects. The majority of the identified problems are grouped into the themes as follows:

Issues related to shared domain knowledge. Due to the lack of knowledge into each other's domain, different competencies and skills, problems arose such as inadequate system documentation obtained by operations, developers' insufficient knowledge of an entire production environment and the whole system integration.

Communication Problems. Because of the poor communication flow between development and operations teams, the knowledge transfer about the new system's working is often missing. Moreover, the communication between these two departments is often very formal through e-mails and documents since they are both separated.

Planning processes and activities. Most of the times, operations teams are not formally involved in the requirement specification and so their non-functional requirements are not considered before the development begins or they will be involved in the system development projects only at the very later stages. Due to this negligence, new systems are put into production very early, before they are complete. Moreover, the process of handing over the new system from Information Systems (IS) development to Information Technology (IT) operations is not very well structured.

Missing Partnership. In order to have a partnership between development and operations teams, there should be a mutual cooperation on goals, risks and benefits. In practice, this alliance is not fulfilled since their goals, priorities and focus areas are different, sometimes conflicting each other.

According to their structured research that involved academic as well as industry-based contributions, França et al. (2016) grouped the motivations behind software organizations' DevOps implementations into following categories:

Organizational structure and policies. The siloed organizational structure that separates IS development and IT operations teams into different departments cause several disadvantages such as an increased developmental cycle time, delayed delivery of software changes, reduced collaboration, lack of trust and increased frustration.

External pressure. Due to increased pressure that demands a faster response to changing requirements, a growing demand for Software-as-a-Service (SaaS) tools requires restructuring in business processes that includes both software development and operations.

Release process. The lack of coordination between development and deployment activities and fear of potential system instabilities that may arise during release processes leads to an increase in the lead-time and thus hinders the continuous value delivery to customers.

Quality demands. In order to balance the lack of developer's knowledge on non-functional features such as maintainability, reliability, and high quality expectations on the highly complex software systems, it is necessary to make changes in the development processes of those systems.

Socio-technical issues. Issues related to social and technical aspects such as cultural differences result in communication issues and a lack of trust among stakeholder groups including development and operations. This may result in long release cycles and low product quality.

Erich et al. (2017) produced a qualitative study on DevOps usage at six different organizations that involve Information System (IS) development. According to their report, the participated organizations intend to achieve various goals such as increase in velocity (i.e., amount of possible work per sprint), increase in product quality, reduce in release time and lead time, support for automation, improvement in problem solving and feedback with the help of DevOps implementation.

3.2.3 Factors associated with DevOps Adoption

A DevOps implementation requires notable effects on processes, products, associated technologies, organizational structures and business practices (Zhu et al. 2016; Wettinger et al. 2014). These factors are separated into human oriented factors and non-human oriented factors according to their characteristics and are discussed further.

Human oriented factors:

There is a general consensus among scholars that the DevOps notion emphasizes on human aspects such as collaboration, trust and empathy (Walls 2013). These human aspects required for a successful DevOps adoption is mentioned as DevOps culture by different authors. A DevOps culture encourages various individual, team-level and organizational changes within a traditional software focused organization. The most mentioned factor is the **collaborative working environment**, where individual members have a shared goal that provides importance towards business value (Huttermann 2012; Walls 2013; Davis and Daniels 2015). Presence of **cohesive teams** that involve people with different skill-sets (for example, development and operations) working together (França et al. 2016). People have an **optimistic and growth mindset**, and believe in continuous learning and improvement (Davis and Daniels 2015). **Open and frequent communication** among stakeholders to discuss, understand and learn from mistakes leading to blamelessness and learning organizations (Davis and Daniels 2015). **Culture of sharing** information and knowledge among team members that helps to bring people with cross-skillsets and prevents organizational knowledge drain (França et al. 2016). **Trust** on individuals as well as teams for example, development team is provided with production environment access because of the confidence on them. At the same time, management should also trust the team members and give them enough freedom that allows for creativity and innovation (Walls 2013; Ravichandran et al. 2016a). **Respect** should be given by everyone to every other individuals and their contributions. At the same time, management should consider all the roles involved in the project with equal respect so that everyone will be considered equally (Walls 2013; Ravichandran et al. 2016a). Practice of **continuous feedback** that appreciates the achievements and encourages to improve shortcomings, but also indicating the misapprehension by being transparent yet constructive (Walls 2013; Swartout 2014). Practice of **aligned incentives and responsibilities** by the management so that the team members will be properly recognized for their work towards successful product that creates business value and satisfies customers (Huttermann 2012; Davis and Daniels 2015). Figure 3.4, adapted from Swartout(2014) represents the various cultural factors required for successful DevOps implementation.

Non-human oriented factors:

One of the commonly perceived benefits of DevOps is faster delivery, which is achieved through the increased focus on process automation (Swartout 2014). DevOps advocates automation in order to reduce manual errors that hamper the software quality (Swartout 2014; Smeds et al. 2015), increase repeatability of tasks (França et al. 2016), remove monotonous work that costs unnecessary human effort (Lwakatare et al. 2015), and also to achieve consistency (Swartout 2014). Although many studies acknowledge automation as a substantial part of DevOps implementation, they are mostly referring to those that are being accomplished with the help of the cloud computing paradigm (Wettinger et al. 2014; Stillwell and Coutinho 2015; Bucena and Kirikova 2017). However, a DevOps implementation is also possible with other computing paradigms (Wettinger et al. 2014). In case of financial organizations such as banks, which are known as 'the slow adopters of technology' (Hon and Millard 2016), a DevOps implementation does not have to be always around cloud solutions.

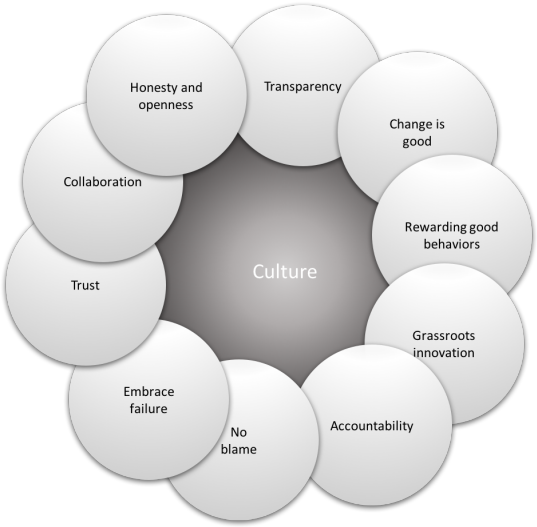

FIGURE 3.4: Interconnectedness of all human factors associated with culture for Continuous Delivery and DevOps adapted from (Swartout 2014, p.58)

This situation needs to be considered well as part of this study since the prospective solution is specific for such financial organizations.

The level of automation embraced by the organizations allows them to adopt a number of continuous software engineering practices such as continuous delivery and continuous monitoring (Davis and Daniels 2015). For example, provision of automation testing tools enables Continuous Testing, which permits the team to test the code as soon as they are written without waiting for the test phase to begin as it happens in traditional development methods (Fitzgerald and Stol 2014). Likewise, an automated testing infrastructure together with a deployment management system, deployment tool, and monitoring infrastructure can help to realize Continuous Deployment, which is the practice of frequently deploying incremental software updates into production (Savor et al. 2016). Other such software engineering practices frequently found within DevOps implementation are: Continuous Integration, which is the frequent integration of newly written code into actual software's existing code (Davis and Daniels 2015) and Continuous Delivery, which is the ability to deploy software at will (Neely and Stolt 2013). Figure 3.5 offers a comprehensive view of all possible continuous software engineering practices among business, development and operations, and it has originally been devised by Fitzgerald and Stol (2014).

As shown in Figure 3.5, Fitzgerald and Stol (2014) involve several other continuous software engineering activities as part of DevOps namely, Continuous Security (i.e. considering security as a key concern throughout the project), Continuous Compliance (i.e. ensuring compliance during the whole development process), Continuous Use (i.e. strategy focusing on customer satisfaction leading to customer retention), Continuous Trust (i.e. developing trust through effective communication), Continuous Run-Time Monitoring (i.e. regular observation of run-time behaviors that aims on early detection of non-functional issues), Continuous Improvement (i.e. achieving incremental advancements with the help of lean approaches) and Continuous Innovation (i.e. conscious adaptation in return to dynamic market conditions).

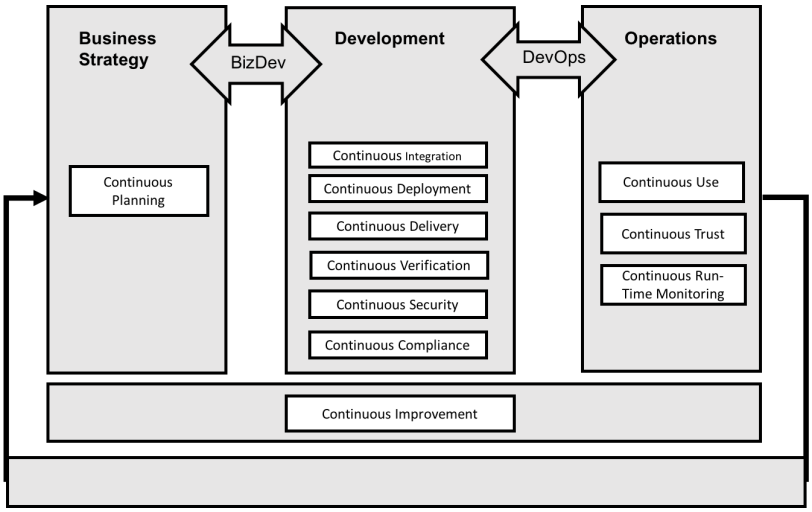

FIGURE 3.5: A holistic view of continuous software engineering practices from Business, Development and Operations adapted from (Fitzgerald and Stol 2014, p.5)

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

TABLE 3.2: DevOps Capabilities and Enablers reproduced from (Smeds et al. 2015, p.171)

DevOps was defined by Smeds et al. (2015) as a set of engineering capabilities reinforced by certain cultural and technological enablers (Smeds et al. 2015). This definition along with the given Table 3.2 elucidates the relationship between the previously discussed human and non-human oriented factors.

In the Table 3.2, capabilities emphasize the continuous nature in the basic software development sub-activities such as, planning, development, testing and deployment. Cultural enablers are the list of human attributes to be possessed by DevOps

teams. Technological enablers highlight the need of automation in all possible activities within software development and delivery so that human errors can be reduced to achieve high quality deliverables. It is interesting to note that continuous planning being considered as part of DevOps whereas Fitzgerald and Stol (2014) mentioned that as a 'BizDev' activity.

3.2.4 DevOps Maturity Models

There are quite a few DevOps specific maturity models available in literature, which are summarized and reviewed in this section in order to enrich the understanding of existence and also the prospective model can be utilizing the best of knowledge obtained from these existing models. Additionally, it is discussed how these models are not appropriate to answer the considered research questions of this study.

DevOps Maturity Model by Mohamed(2015)

The first one is the DevOps maturity model developed by Mohamed (2015), which is based on the renowned software process maturity framework, CMMI maturity model. Mohamed (2015)'s model has 5 maturity levels, whereas each level is defining certain maturity criteria in terms of 4 dimensions namely Quality, Automation, Communication/collaboration and Governance. The maturity model that can be seen at Figure 3.6, is organized in such a way that the initial level defines the ab-

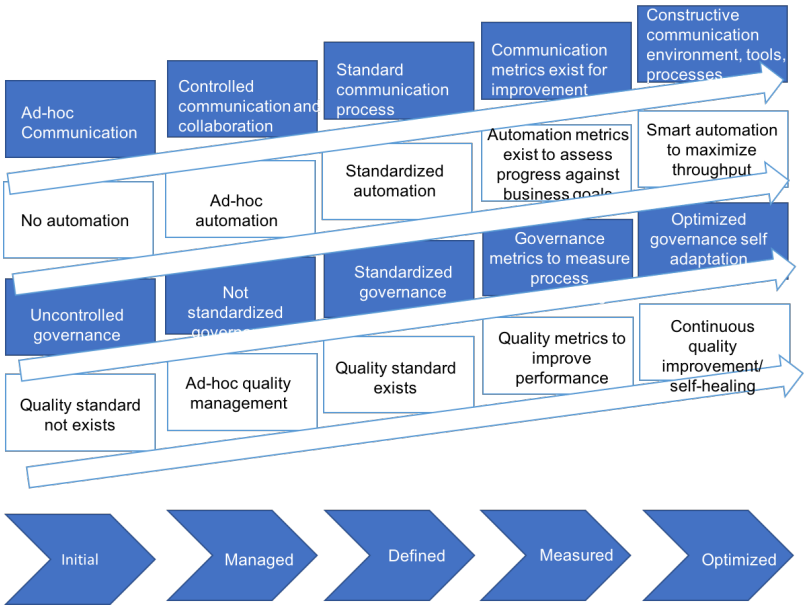

FIGURE 3.6: DevOps maturity model adapted from (Mohamed 2016, p.26))

sence of a controlled process with regards to each of the given dimensions whereas, the final level defines the presence of process that focuses on betterment of those dimensions. For example, the first dimension, communication is characterized with no clear processes/tools/roles that allow effective communication among stakeholders and centralized decision making by one role is often the case. Successive maturity

levels are defined with characteristics of incremental changes required by the organization to improve communication among various stakeholders. The final level of communication dimension specifies that an organization already has the essentials in place and their further focus is on enhancement of the same according to the organizational goals. The second dimension, automation encourages organizations towards robust automation framework aligned with organization's objectives. The third and fourth dimensions emphasize respectively on specifying clear governance processes and quality assurance activities with proper tools and processes while measuring them with appropriate metrics.
Although this model's dimensions focus on the foremost aspects involved DevOps implementation, this does not include many of the essential human and non-human oriented factors addressed in subsection 3.2.3. Moreover this model offers a generic framework rather than a industry-specific framework, which is the focus of the current study. Additionally, this model does not allow organizations to tailor their own implementation approach since the levels are fixed to certain capabilities of every dimension.

DevOps Maturity Model by Bucena and Kirikova(2017)

Bucena and Kirikova (2017) have proposed a DevOps maturity model which follows Capability Maturity Model's 5 maturity levels viz. Initial (Level 1), Repeatable (Level 2), Defined (Level 3), Managed (Level 4) and Optimized (Level 5). All these

Area	ID	Initial (Level 1)	Repeatable (Level 2)	Defined (Level 3)	Managed (Level 4)	Optimized (Level 5)
People	P1	Teams organized around skillsets	Team organized around deliveries	Team organized around projects	Team organized around products/ business lines	Interdisciplinary teams organized around KPIs
	P2	Ad-hoc learning	Team learning	Value stream learning	X-process learning	External learning
	P3	Ad-hoc approach regarding competences development	Competences are developed with the help of training and development	Analysis of exiting competences and future development	Mentor usage	Continuous capability improvement
Culture	C1	Restricted Communication	Rapid intra-team communication	Rapid inter-team communication	Frequent collaborative communication	Rapid feedback
	C2	Uncommunicated vision	Clear delivery requirements	Clear project requirements	Clear product/business line requirements	Clear organization requirements
	C3	Lack of cultural awareness	Awareness of cultural aspects that may help or hinder day-to-day business	Cultural traits that support business strategies have been identified	Culture viewed as an asset to be managed	Desired elements of the culture are identified, ingrained and sustainable, thus creating "the way we work here"
	C4	Poor, ad-hoc communication and coordination	Managed communication	Active collaboration	Collaboration on based on process measurement, which allows to identify bottlenecks and inefficiencies	-
	C5	Sub-innovating / no innovations	Innovations by necessity	Innovation by design	Strategic innovation	-

FIGURE 3.7: A sample fragment from the DevOps maturity model of (Bucena and Kirikova 2017, p.9)

maturity levels are defined with respect to four enterprise areas namely Technology, Process, People and Culture. Each of these areas are having multiple rows whereas each of them is identified with a specific ID such as T1 (i.e. Technology 1), P2 (i.e. People 2) and PR3 (i.e. Process 3). In this way, Technology area appears with 9 rows, Process area depicts 6 rows, People and Culture areas are illustrating 3 rows and 5 rows respectively. Each specific 'chunk' (as mentioned by the authors) in the technology area (Levels 2 to 5) can be associated with their corresponding DevOps practices

given in a separate table. This is definitely a good inspiration model as the authors have considered a good number of details for both people and non-people aspects involved in DevOps implementation. On the other hand, this model does not satisfy the research questions considered in this study since this model does not consider Agile principles and practices as much it does with DevOps practices. However, it does mention 'Agile Development' as one of the values (PR2 - Level 3) within the model but not more than that. Moreover, this model enacts a very generic solution, not specific to any type of industries and so it does not consider the limitations of any particular industry. Due to the space limitations, an excerpt of the model has been depicted in Figure 3.7.

DevOps Maturity Model by Feijter et al. (2018)

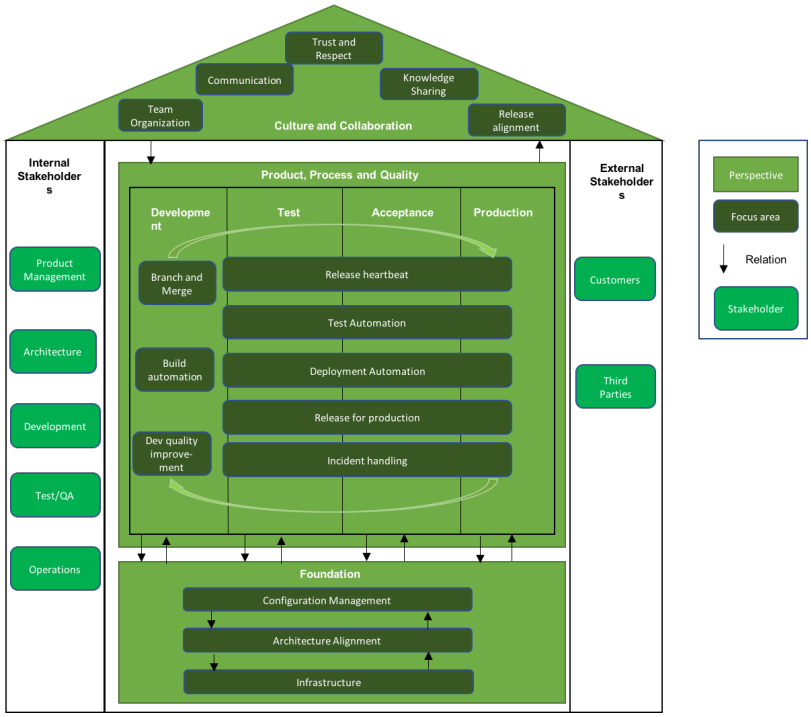

FIGURE 3.8: DevOps Competence Model adapted from (Feijter et al. 2018, p.248)

The next one to be discussed is a DevOps competence model and its corresponding maturity model developed by Feijter et al. (2018). These models are specific for Software Product Organizations (a.k.a. SPO), which are organizations that produce standard software(s) to be used by several customers and so they are different from organizations that produce customized software for a specific customer (De Weerd et al. 2006). The competence model shown in Figure 3.8 portrays the various focus areas within the specific perspectives, their relationships and the important stakeholders involved. The focus areas are grouped into perspectives namely 'Culture and Collaboration', 'Product, Process and Quality' and 'Foundation' in a way that both people and non-people related factors are considered within the model. Each of the mentioned focus areas are identified with several capabilities, which are positioned into the maturity matrix (Figure 3.8) based on collected data from literature as well as practitioners. The maturity model shown in Figure 3.9 contains all the

perspectives, focus areas within each of the perspectives and rank-ordered capabilities related to every perspective. While the competence model shows all the aspects related to DevOps implementation, the maturity model will be of help to measure the current maturity level and to show the incremental path towards attaining the highest maturity possible.

Focus area/level	0	1	2	3	4	5	6	7	8	9	10
Culture and Collaboration											
Communication		A				B	C			D	E
Knowledge sharing				A		B	C				D
Trust and respect						A	B	C			
Team Organization		A	B						C	D	
Release Alignment				A					B	C	
Product, Process and Quality											
Release heartbeat		A				B	C		D	E	F
Branch and merge			A	B		C		D			
Build automation			A	B		C					
Development quality improvement			A			B		C	D	E	
Test automation				A	B	C			D		E
Deployment automation					A	B		C			D
Release for production					A			B	C	D	
Incident handling			A					B	C	D	
Foundation											
Configuration Management		A	B		C						
Architecture alignment		A						B			
Infrastructure			A				B	C	D		

FIGURE 3.9: DevOps Maturity Model adapted from (Feijter et al. 2018, p.251)

The current study is stimulated by the aforementioned work of Feijter et al. (2018) and so considers to create similar artifacts such as a competence model and a maturity model that offers an incremental approach. However, the current study will be different from the discussed work of (Feijter et al. 2018) in the following ways: the current study concentrates on financial organizations whereas (Feijter et al. 2018) focused on SPOs; the current study extensively considers both Agile and DevOps principles and practices but the work of (Feijter et al. 2018) deals with DevOps only. Furthermore, the current study extends the work of (Feijter et al. 2018) by applying the models to measure the present maturity level of certain organizations and then compare it with performance metrics specific for Agile and DevOps environment.

3.2.5 Relationship between Agile and DevOps

According to our analysis on Agile and DevOps, there is quite some uniformity between the subjects can be identified. Although the characteristics are associated, yet the relationship between Agile and DevOps has not been studied extensively. On the other hand, Lwakatare et al. 2016 highlight that there are several multi-vocal literature studies discussed DevOps in relation with Agile, Lean and Continuous Delivery (CD) in comparison with a small number of scientific articles. As per their study, Agile organizations find it easy with DevOps adoption whereas, Waterfall

practices are very much incompatible with DevOps practices and so DevOps adoption is not easy for them. In case of implementation, DevOps reinforces Agile software development practices and roles and so improvements in the usage of Agile methods such as Scrum. In terms of goals and values, both Agile and DevOps try to break down organizational silos while encouraging collaborative working, focusing on rapid and incremental releases, quick feedback and correction.

The framework that covers the Agile and DevOps practices in the context of Systems Engineering practices backing the statements of Lwakatare et al. (2016). Systems Engineering (SE) processes are those that deal with cost effective development of high quality systems (Geurts 2016). In Figure 3.10, Geurts (2016) indicated the Agile and DevOps practices used in SE processes and in addition, he highlights the DevOps practices that extend Agile practices.

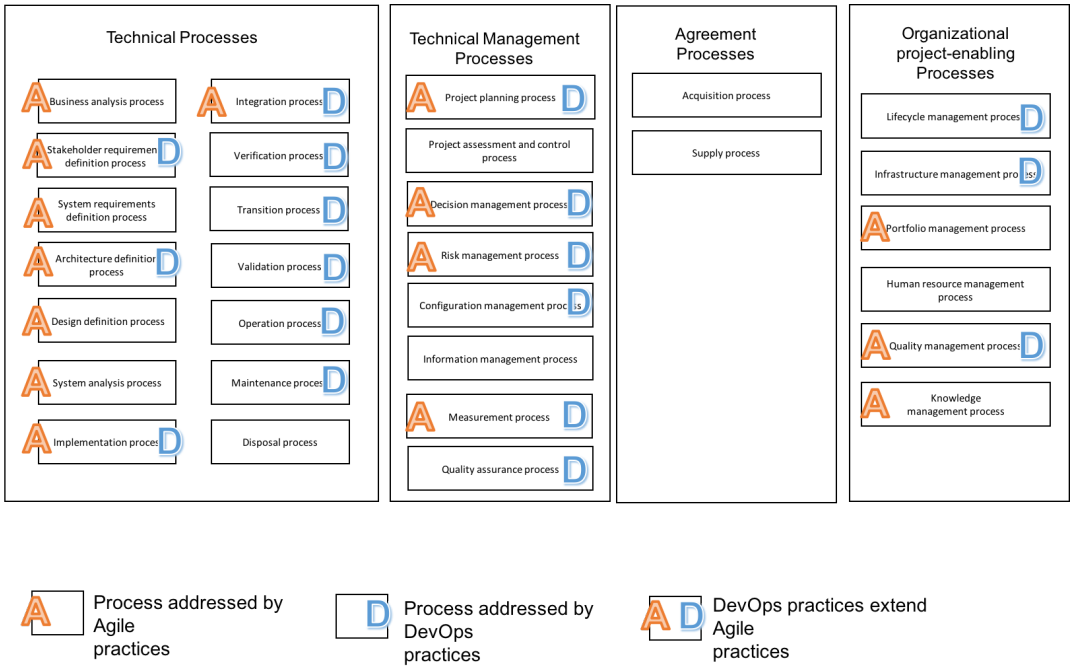

FIGURE 3.10: Coverage of Agile practices and their DevOps extensions in the context of Systems Engineering adapted from (Geurts 2016, p.105)

As a result of their grounded theory analysis on modern web application development, Bang et al. (2013) explored several DevOps related concepts and categorized them according to the related knowledge and perspectives. As a consequence, there resulted four DevOps related perspectives namely collaboration, automation, measurement and sharing. Interestingly, the authors highlighted that the Agile software development methodology supports all the identified perspectives of DevOps. Jabbari et al. (2016) also summarized the studies that relate to DevOps and Agile, and he highlighted that several other authors agree that Agile enables and supports DevOps practices whereas DevOps extends existing Agile practices.

3.3 Performance Metrics

In the software engineering context, metrics are important irrespective of the development methodology which is being followed within an organization. Metrics help to control and improve what is being done and how is it being done (Kupiainen et al. 2015). The metrics should be carefully selected so that they can demonstrate how their work impacts the business of an organization (Ravichandran et al. 2016a). In this section, we review the Agile and DevOps metrics from the existing literature in order to get a comprehential knowledge on the subject.

3.3.1 Agile Metrics

Kupiainen et al. (2015) performed a systematic literature review on the metrics used in Agile software development environment. The authors did not only collect the metrics from the existing literature but also categorized them based on their purpose of usage, ranked them according to their number of occurrences and perceived importance factors from past studies, and lastly, mapped their findings to Agile principles.
According to Kupiainen et al. (2015), Velocity (i.e. number of estimated user stories vs number of actual completion of user stories (Javdani et al. 2001)) is the highly ranked metric with maximum number of occurrences and a high importance factor out of 25 such metrics. The other such highly influential metrics are effort estimate (i.e. estimated effort per story), customer satisfaction, build status (i.e. build status as broken or not), technical debt and progress as working code. The authors identified that the found metrics are used for one of the following reasons: Sprint and project planning, sprint and project progress tracking, understanding and improving quality, fixing software process problems, and motivating people.

3.3.2 DevOps Metrics

With respect to DevOps specific metrics, there are very few studies found to list them out. Regarding this subject, Ravichandran et al. (2016), provides some useful advice that will be taken into account during the performance metrics part of this study. The authors advises that the metrics should not be into one of the following problematic classes:

- **Vanity Metrics**: Metrics that accolade and encourage an erraneous behaviour within the team such as 'lines of code produced'.
- **Intra-team Metrics**: Metrics that disturb the cross-cultural collaborative behaviour within the DevOps teams. For example, the operational effectiveness metrics that affect the number of deployments.
- **Traditional Metrics**: Metrics that are traditionally being within the organization but not really useful with the new ways of working, for example: Mean-time-between-failure, which is not useful in the faster delivery focused environment.

Knowing what type of metrics should not be considered to measure the DevOps effectiveness, Ravichandran et al. (2016) points out that the metrics should measure

people, who are the main change indicators. Additionally, it is also important to observe how the changes made in the existing processes and technology is affecting the success path.
Ravichandran et al. (2016) proposes the following characteristics to be possessed by metrics that are being considered: obtainable (i.e. the metrics are possible to collect), reviewable (i.e. the metrics should be thoroughly reviewed), incorruptible (i.e. the metrics is not biased to certain sub-group of the team), and actionable (i.e. the metrics supports improved decision making). These characteristics will be considered during the performance metrics identification part of this study.
According to their research survey conducted in 2014, Elliot (2015) revealed the following DevOps specific business metrics obtained from 20+ Fortune 1000 organizations.

- **Productivity related metrics:** Speed, velocity, impact analysis, build and test automation, configuration automation, and time to market
- **Quality related metrics:** Availability, requirements analysis, business stakeholder involvement and support, metrics that help to identify issues earlier through continuous testing and integration
- **Operating expense related metrics:** Cost optimization, cost modeling, metrics that encourage fail fast and fail cheap, allocation of IT
- **Capital expense related metrics:** Utilization, cloud-based systems, and convergence

The current study identifies the drivers behind financial organization's DevOps implementations and also their performance metrics, so that the relation between those two can be established. Additionally, the current study aims to identify a generic list of performance metrics that are useful for Agile based DevOps environments.

Summary of literature review

The current literature study reviews the relevant literature in order to collect the obligatory knowledge required for the execution of the current research. Firstly, section 2.1 surveys the Agile methodology to understand the rationale behind organization's will towards Agile adoption, various factors that get affected with Agile transformation, and the essence of existing Agile related maturity models. Secondly, section 2.2 revises the DevOps related concepts to grasp the perception behind the term 'DevOps' to discern the claims towards organizations' DevOps move and the involved aspects of DevOps implementation. Furthermore, section 2.2.4 analyses the existing DevOps maturity models to identify the perspectives of other authors who performed works similar to this study. Section 2.2.3 brings out the relationship between the concepts of Agile and DevOps found from literature. Finally, section 2.2.5 reviews the performance metrics in light of Agile and DevOps separately.

4 Current Agile and DevOps practices at Dutch Banks

In this Chapter, we present the summary of the data collected from the practitioners from financial software organizations. Since the type of financial software organizations is diverse, due to the time restrictions, the data has been collected from people working at 7 different sub-organizations ie. departments from 2 large banks in the Netherlands. The data has been collected through semi-structured interviews as mentioned in Section 3.2.2. The interview questionnaire (Appendix A) has been designed based on the literature collected data and so the classification of adoption factors was kept as such in order to correctly interpret the data during the analysis. Since most of the sub-organizations are already following Agile and DevOps, the current data could not be largely distinguished to mention that it was due to either Agile or DevOps adoption. Thus, the current chapter summarizes the factors without differentiating between Agile and DevOps. Moreover, both the organizations have chosen Scrum as their Agile method and so only the Scrum practices are discussed throughout.

4.1 Factors associated with Agile and DevOps adoption

4.1.1 The organizational factors of Agile and DevOps adoption:

In this classification, the main objective was to understand the changes observed in the culture and the environment of the organization. Additionally, the factors such as organization structure and the importance of management buy-in were also discussed during the interviews.
First of all, both the banks have brought up DevOps teams by combining both the developers, operations engineers and other roles such as testers, business analysts in one team (iv-pi-6 and iv-pi-4). This setup has already brought more transparency in the workplace, more awareness about each other and open communication that helps to solve the problems faster than earlier (iv-pi-6). However, the team members are still encouraged to broaden their skill set to get more T-shaped by given with more freedom and resources such as training (iv-pi-7 and iv-pi-4). One of the interviewees talked about their new team setups as follows:

"*there is no more like Dev team and an Ops team.. we made a single DevOps team with 3 or 4 Dev Engineers.. and like a couple of Ops engineers.. and Ops engineers are completely involved in everyday standups, they know exactly what is coming in the next sprint, what is going live..and also if there is an issue in the production, it is not about like Ops only has to take care.. but as a team you have to handle the issue.. so if there is an issue, as a team we*

discuss, I [as a developer] am more actively involved in the after-care issue... so that is the main change.." (iv-pi-5)

In terms of the Agile and DevOps, both the organizations are strategically moving forward to them as decided by the board. However, when it comes to the level of adoption by the people within the organization, they are supposed to follow this new way of working as that is the way the organization is stepping towards. One of the interviewees proclaimed that the transformation should be top-down, starting from the higher level of management since the transformation will not be successful without the management buy-in (iv-pi-6). Thus the people at the management should be properly educated about the transformation before going to the teams (iv-pi-6). For this, Agile coaches, professionals who help with the transformations, are temporarily hired by both the organizations. They help the management people by having one-on-one sessions to change their traditional management style to more supportive Agile-friendly style (iv-pi-6, iv-pi-7, iv-pi-1 and iv-pi-2). One of the interviewees explained the change in management as follows:

"At first, they [management] did not know it well either because they have not changed their mindset as well. Because from a manager's perspective, going from a project management into Agile/Scrum is kind of a tricky change. So for that part, we had agile coaches for [a certain period] during the first [x number of] years of the implementation, who came here as well with the teams to show us in practice how to do things[...] I would say that the management has really changed the cold way of thinking of the teams, whether that is on the evaluation criteria for each employee, how the team is perceived as successful or not, they have managed more or less to make it compatible with Agile. So being a successful team, also goes to being a successful agile team and implementing successful scrum." (iv-pi-7)

With regards to the organization structure, traditionally both the organizations had the hierarchical structure where teams used to be managed by a single person for their delivery, skills, and performance. However, after the transformation, both the organizations are following a structure where the Product Owner is being responsible for achieving the maximum value from team's work; and IT management is responsible for the human resources part including the skills and performance management (iv-pi-6 and iv-pi-2). Moreover, in the earlier hierarchical structure, there were no restrictions on the team size and so every x number of teams were governed by a lead team member (example, lead developer or lead tester), in teams that are too big to be managed only by a Project Manager (iv-pi-1). However, now the teams are restricted in size to have a maximum of 10 members and they are also becoming more self-steering in terms of being responsible to plan and manage their own work (iv-pi-1, iv-pi-5 and iv-pi-7). Additionally, they are involved in performance assessment of peers in the name of 360deg feedback and sometimes new hirings as well (iv-pi-7). One of the interviewees explained the 360-degree feedback session as follows:

"That [360 degrees feedback] means that you ask multiple colleagues working with you, not only your own team members but also stakeholders from the outside, people you interact with, you ask them for their perspective on your performance on certain items that you address at [a specific time] of the year" (iv-pi-4)

In the matter of the corporate culture, several interviewees noted that the new way

of working has brought openness to their corporate culture (iv-pi-6). People are encouraged to discuss unclear things with each other and also with the management and business, to provide and accept feedback constructively, to work as a team and be responsible for their work as a team. One of the interviewees mentioned that their performance evaluation includes a component that assesses their work as a team and gets their evaluation score as a team and thus everyone would feel responsible towards their teamwork (iv-pi-5). Another interviewee mentioned that the organization's focus was only on the delivery earlier, but now it includes delivery as well as continuous improvement of people and process (iv-pi-1). For example, the interviewee-1 described his observation on the cultural change as follows:

"What we see is that there are more discussion within the teams.. also the retrospective, after the sprint.. we see changes in the culture.. they are more talking to each other than the old days.. in the old days, it was more 'I am doing my thing and throwing it over the wall' but the other one [Agile] is letting people more into discussion... " (iv-pi-1)

Many of the interviewees welcomed the freedom towards their skill development that they are getting after the organizational transformation. The team members are encouraged to become more cross-skilled, and the organization enables that by allowing them to choose their interesting career path and encourage them to set their own goals in-line with the organizational goals (iv-pi-1 and iv-pi-3). People are given time and resources like internal and external training to realize their goals in order to expand their skill set.

4.1.2 People factors in Agile and DevOps adoption:

The second classification concentrated on understanding the transformation of people in the teams, their relationship with the customers and the organization's role on this part.
In case of both the banks, the IT departments are working for their business counterparts who are considered as their customers in terms of bringing work to them. Thus the deliverables are mostly custom software or services which satisfy the needs of those business counterparts. Moreover, the IT departments consist of people who are direct employees of the bank (aka. Internals or internal employees) and also externals (aka. external employees who work temporarily on project basis) who are brought up to work temporarily within various projects. This is important to know since the external employees might not be directly given with the opportunities that of internal employees for example, trainings. However, the case of external people is not considered for this research.
"Experience does not [always] matter [...] they [the organization / management] are looking for people to be open-minded in a way that you are not locked in a way of working that you know and you think that only works [...] people wanted to know that this person [new candidate] can operate as a team member and as an individual person. Because scrum teams are self-organized so we don't need someone who knows what they are doing their work, but someone who needs to take ownership of whatever they are building and the team is building, and so he can advise and suggest how something can be improved, especially due to the continuous feedback nature of the Agile /Scrum way of working. [...] so in some ways you want people to take ownership of what they are doing and be open-minded to try out new things and how to do it. Because we are still learning also and it is not that we have implemented and we know it. Scrum is not just one solution but you have to adapt and make it work for

your specific situation. So you need people that are open to that and also can contribute to that." (iv-pi-7) Several interviewees asserted that open communication is the most important characteristic of an agile team as well as team members. By discussing details openly, without just assuming or delaying information or thoughts, giving and accepting feedback constructively, and by sharing knowledge with others, a person is considered to be a good team member within an ideal agile team environment (iv-pi-3, iv-pi-1, iv-pi-5, and iv-pi-4). Additionally, as quoted from (iv-pi-1), people are expected to take ownership of the tasks and just not be confined to their job descriptions (iv-pi-1 and iv-pi-3). Thus the transformation to happen within the people mindset can be described as 'me to WE'. Agile coaches are used to making this transformation within the people factor of both the involved banks.

Before Agile way of working was introduced, the teams had very less contact with their business counterparts (aka. Customers in this context) and it was mostly during pre-development and post-delivery of the complete project or a big part of it which usually lasts for months. However, thanks to Agile, this time has really been shortened, and the role 'Product Owner', who is usually from the business, is constantly in touch with the teams (iv-pi-5). Although many teams have a separate Product Owner for every Scrum team, interviewees told of situations where several teams having one Product Owner as well. In the latter situation, there used to be another bridge between the Product Owner and the teams. However, in all the situations, the teams have the freedom to contact their Product Owner at any time for any clarifications. This situation has been told to bring in a big change in their way of working since the Product Owner concentrates only on the business value achieved from the team's work but not the people themselves. Additionally, the teams get to meet the rest of the business team as well during the Scrum events such as Sprint Review meetings, which happen at the end of every sprint. This satisfies the need to have a direct communication with the business in a regular manner (iv-pi-7, iv-pi-6, iv-pi-3 and iv-pi-2). The iv-pi-1 gave some light on the relationship among the PO, team and the IT side management as follows:

"*He [The Product Owner] sits next to the team and he is representing the business [...] basically the PO says what I [the business] want.. the team has to find out how they can achieve that and come back...[...]but the PO does not say that this is the way you should do it.. the team decides how they want to do it [...]in case, they are not able to give what they arrive into, then [the corresponding role from IT side management] is there.. he will step in and see how to optimize, how to make it happen... he [The Product Owner] will only give the priority and ask the team to catch up with it.. the [the corresponding role from IT side management] should enable that..*" (iv-pi-6)
Altogether, the interviewees mentioned that People factor are the most important as well as the most difficult one to be transformed in order to realize the real benefits. People are expected to become cross-skilled, adaptive, responsible and fertile minded.

4.1.3 Process factors of Agile and DevOps adoption:

As mentioned in the People factor classification, people working at the teams may include both internal and external employees. Therefore, there could be differences in the processes undergone by the people within banks since externals might not go through the same processes as that of internals, for example, their hiring process

and performance evaluation process might be different. Since all the interviewees participated in the interviews were internal employees, the processes imposed on external employees are neither included nor considered for the research.

This dimension concerns about the complete software development and delivery process including change management, and the project management process which is more about keeping the project status under control.

During the project initiation, the most high-level business requirements called 'epics' are brought into the corresponding IT sub-organizations by the business organization. Every 'epic' comes with several 'features' and every feature is broken into 'user stories' where epics are the most high level and the user stories are most low-level requirements. Once the stories are established, they will be prioritized by the PO, refined by the team (with or without the PO), built, tested, and finally deployed by the team.

With the traditional way of working, everything was well documented at the beginning before the actual software development starts and so every change request was always going through a separate extensive process before being accepted. But now, the actual build is broken into many small pieces which can be developed within a sprint of 2 or 4 weeks long and so every requirement is allowed to be changed before it is taken into the sprint. This is such a remarkable freedom provided by Agile which lead to high user satisfaction. The banks felt advantageous with this flexibility as like any other different organizations. However, one of the interviewees confessed that it is quite not feasible to have the unstable requirements until the last minute. Therefore, they will define all the epics and features for one quarter and the corresponding user stories may change but not the top level requirements unless the changes are crucial. Moreover, changing the user stories until the last moment is also not efficient as they were leading to improper refinements, and imposing a lot of rework. Thus, some interviewees mentioned of having a clear overview of upcoming 5-6 sprints and the changes in them should be considered well (iv-pi-3 and iv-pi-6), which also helps to make the necessary arrangements about anything required from different teams. The interviewee 6 described this upcoming sprint outlook planning done by the Product Owners as below:

"Epics are defined already [...] the feature is held by the PO.. so he knows what feature needs to be delivered.. so lets say, you have 5 or 10 POs and they all sit together and say 'Okay, this 90 days, we have to deliver so many features' and for these 3[features/stories], we have dependencies.. 'okay, you [Product Owner -1] have 5 stories under this feature and I [Product Owner -2] have 3.' so that high level planning, they [Product Owners] do.. then they take the user stories back to the team, then do an estimation to see whether they can fit into their timebox.. if it doesn't fit, then come back into their PO meeting again to say that, 'I [Product Owner-2] can not finish all the 3 but I can finish only 2..' but then if the clarity is not there, they again put it in on the table saying that, 'this feature can not be done because ABC is not very clear..' [...] so one or two iterations, that keep happening.. before which, it [features/ stories] really becomes ready for a team to burn in the sprint.. in spite of doing all these, you will find some uncertainty, but then uncertainty is in very small extent of it.. so this is the process that happens continuously on a bi-weekly basis.. looking forward to the next 6 weeks.." (iv-pi-6)

Although this outlook for several sprints is necessary, some interviewees agreed that

they should be considering the changes imposed in the upcoming sprints or even an ongoing sprint based on the severity of the change and some others said that they will never allow any changes within the ongoing sprint. However, all mentioned that a discussion is always necessary in any such cases (iv-pi-4, iv-pi-6 and iv-pi-7). The interviewee 6 mentioned the advantage of following Agile processes as follows:

"the information, the good news, and the bad news is all delivered in the bi-weekly basis [at the end of sprint] but not at the end of the project.. that is the bigger change" (iv-pi-6)

For managing the projects in an Agile way, Scrum offers several events such as daily stand-ups, where team updates on their individual progress and discusses the impediments; refinements where the dependencies are identified; planning meetings where the actual work can be planned based on the availability of resources and dependencies; sprint review meetings, where the stakeholders are demonstrated with the last sprint's shippable product and discuss it with the team. The interviewee 3 and interviewee 7 expressed their opinion about the daily stand-ups and review meeting as follows:

"It (Daily stand-up) makes it easier to grab on to [the ongoing sprint work].. you have more feeling with the process itself instead of guessing the process or accepting the stuff someone else says.. Now you can see and experiment it yourselves.." (iv-pi-3)

"[...]in the scrum review meeting where every stakeholder will come and see what we developed in the last sprint, so they [the stakeholders] can the feedback and to see what can be done in the upcoming sprints. So, these kind of meetings were really helpful in the way that we needed to understand what the business wanted from us [IT team], and they [the business team] could understand what we can give them" (iv-pi-7)

Since the above meetings are meant only within teams or departments, the organizations have to find their own ways to resolve the cross-dependencies between the teams and departments. Principally, because of their self-steering nature, teams are expected to resolve these dependencies on their own (with the help of the management if required) in a timely manner, so that the project status will not be affected by any delays. However, the interdependencies between the systems can be complex, and the number of involved teams can be very large at times. Thus, it may not always be feasible to solve these issues without management support by the teams. As another solution to this dependency issue, some of the interviewees talked about having the 'feature teams', which is a team formed with the people who are skilled working in different systems, united together in work on a single feature or epic. They believe that this formation will resolve the dependency issues that are currently being a present with the current 'component teams', where people are organized as such they are all working for a single system (iv-pi-2, iv-pi-5, iv-pi-6 and iv-pi-7). The interviewee 7 narrated about their way of handling the mentioned inter-dependencies as follows:

"[...]besides the scrum meetings, we have introduced our own meetings to adapt Scrum into our way of working in the sense that, not scrum of scrums, but intra-team meetings.. because our projects are always linked. There is no team that is doing only what they are doing, so you always have dependencies with other teams. So to solve that, we have alignment meetings. Each week, each month, depending on what the project's nature is. We are always aware of each others' work, and when something needs to be in place. If this does not happen,

we always give and get the immediate feedback as soon as we know of it" (iv-pi-7)

The other important factor evolved from the interviews was knowledge sharing. Since most of the scrum teams are having similar setup (Scrum Master, Team, Product Owner etc.) and since the DevOps has broken the organization based on skill sets (Ops team, testing team etc.,), it is important for the people to share knowledge of best practices among each other in order to avoid re-engineering. The Spotify term used to mention this knowledge sharing groups as 'guilds' and many of the interviewees agreed that this knowledge sharing part is given with less importance by the people than it should be.

Overall, most of the interviewees agreed that many of the processes are clear to understand but the difficulty is only in making everyone following them to obtain the full benefits out of it. They mentioned that the processes are required to be adapted in the way they are being followed rather than changing the process itself.

4.1.4 Technology factor of Agile and DevOps adoption:

The technology dimension was intended to learn how the technology intervenes and eases out the processes being followed within an organization.

All the interviewees agreed that automation is significant especially, in their DevOps way of working. By looking at the activities within the Agile software development and delivery processes, almost everyone agreed that the test automation is the first thing that they would like to accomplish and the next one is automatic deployment. When it comes to the continuous software engineering practices, Continuous integration (the frequent integration of newly written code into actual software's existing code (Davis and Daniels, 2015) is something that everyone is aiming to achieve if not done already. Next to that, people are interested to establish the pipeline for the continuous delivery (Continuous Delivery is the ability to deploy the software at will (Neely and Stolt, 2013)). However, when it comes about continuous deployment (frequent deployment of incremental software update into production (Savor et al., 2016)), some of the interviewees were not very sure about having it since they mentioned that the banking systems and their interdependencies are complex.

The interviewees mentioned that the younger technologies are easier to be automated than the older ones, not mainly because of the technology itself, but mainly due to the people working with those technologies. Many accepted that the younger minds have the drive towards automation than those who are working without automation for decades. However, interviewees mentioned that the popularity of cloud technology is getting high with the automation of banking systems as well. The organizations have a separate team which keep discovering the latest tools that enable automation within their organization, and their compatibility with the existing systems or applications. One of the interviewees talked about such an initiative applied within their organization to ease out the automation as follows:

"in our organization, they have a central team who does all the research and the development on the pulling [continuous integration] pipeline actually.. they make sure that the continuous integration happens.. that is for the build.. and continuous deployment happens.. so they are two separate pipelines based on separate technology lines basically.. for java, you

have separate pipeline.. for .net separate pipelines..etc. and they provide the services.. so every team that gives built-in, spot and keeps that built-in and [...] [IT team management personnel] will tell them[the team] to subscribe to that particular pipeline [...] the team will be extending and utilizing the pipeline for their project.. but the central framework, guideline, everything they [the central team] will give.. [...] Maybe not all the technologies are covered by them.. they are given for the popular ones.. for some, it is not there.. then that has to be done by the teams themselves and give it back to the central team.." (iv-pi-6)

4.2 Summary

This summary highlights the important points from the collected data according to the organizational, people, process and technology factors.
Organizational factors:

- Organizational restructuring happened at both banks including both at the team levels and management levels.
- In the new organizational structure, the Product Owner role has the responsibility of achieving business value from team's work; IT management has the responsibility of the human resources part including skills and performance; Agile coaches help to instill the agile principles and tailor the practices within the organizations. The communication between the development and operations have become less formal although various DevOps team structures have been identified.
- The people evaluation methods are revised and 360degree feedback sessions are identified.
- The organizational culture has become less hierarchical and the management teams are becoming less bureaucratic.

People factors:

- Open communication is identified as the most important characteristic required for DevOps environment.
- Agile coaches are found to be helpful with making the transformation among people.
- People within the team are getting more freedom and more authority over the way of performing the assigned tasks of the team. This was perceived as the major difference from traditional way of working.

Process factors:

- All the interviewees follow Scrum method within their teams. Most of them follow all the Scrum defined processes but the processes are adjusted to suit their environment and needs.
- The large scale agile methods have been identified and followed so that the Agile can be followed within the enterprise. No specific name of such methods have been mentioned by the interviewees.

- The requirements are managed as 'Epics - Features - User stories' and they are continuously defined, prioritized and analyzed before the developmental activities by the teams and the business stakeholders ie. Product Owner.
- Various change management activities were identified but no clear organizational level strategies had been heard.
- Knowledge management was identified as an important aspect within the new organizational structure. The 'guilds' are identified as the common practice to share knowledge among people who share the same expertise. However, this was found to be a non-strict activity and so taken for granted.

Technology factors:

- Every interviewee agreed that automation is significant for the new DevOps way of working.
- Younger technologies are easier to automate due to available tools and the younger mind-sets working for those technologies and it is opposite to the older technologies and so legacy systems.
- Cloud adoption is becoming prominent among banks.

5 Towards the development of artifacts

This chapter is presenting the results of data analysis performed on the data collected from both practitioners and the available literature. Based on the results, the basic components namely, drivers, perspectives and focus areas are identified. Later, by grouping the discerned components, the intended deliverables namely, drivers, DevOps implementation framework, and the framework variations per driver are developed. According to the definition from oxford dictionary, "artifact is an observation in a scientific investigation that is not naturally present but occurs as a result of the investigative procedure". Considering this definition, the mentioned deliverables of this thesis are called as artifacts, since they are the conceptual elements which result from our interpretation of the collected data.

5.1 Artifact 1: Main drivers behind Agile and DevOps adoption at financial organizations

The first artifact is presenting the list of drivers, which represent the main reasons why the financial organizations desire to implement DevOps besides Agile. Such reasons are collected in general from the past literature but this list is developed by filtering out the specific motivations of participating organizations.

This artifact is developed since understanding the purpose of DevOps and Agile implementations may help us to bring up the suitable framework addressing the right needs.

5.1.1 Driver 1: Agility and Customer-centricity

The first and foremost driver for DevOps adoption was identified as the 'Agility and Customer-Centricity' and, as previously discussed in section 2.2.4, earlier studies have identified that DevOps supports and enhances the Agile practices (Lwakatare et al. 2016b). Most of the agile practices are corroborating to this main driver as they are allowing the organization to be more flexible to the customer requirements by being constantly in communication with them (Diebold and Dahlem 2014). Tseng and Lin (2011) defines agility as the enterprise's ability to respond fast to the changes in the market and customer demands (p.3697). Liang and Tanniru (2007) defines customer centric information systems are those that consider customer value as the main business driver and they are dynamically developed according to the customer preferences and needs. From the interviews, it was understood that the financial organizations are moving towards Agile and Devops in order to develop the customer

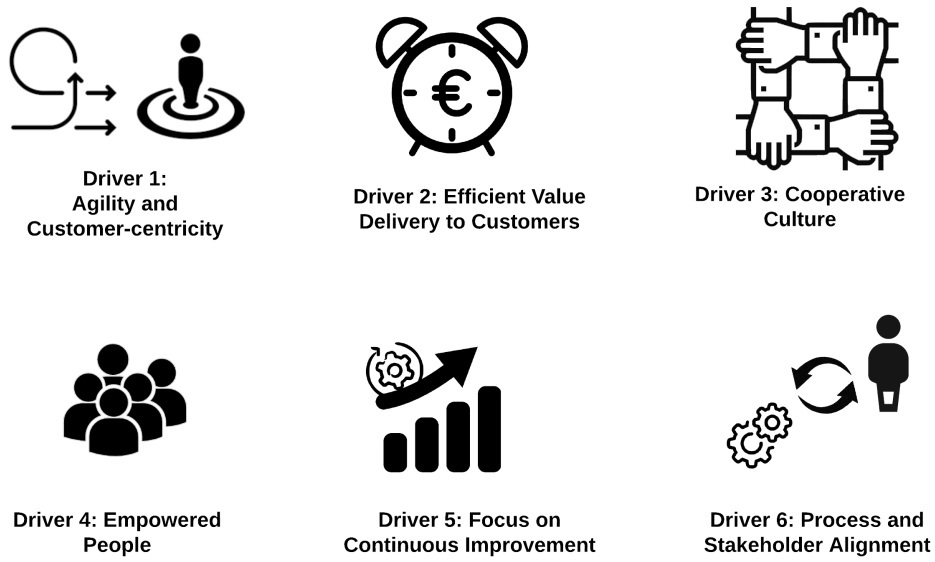

FIGURE 5.1: Drivers for Agile and DevOps adoption at Dutch Banks

centric software products while meeting the internal requirements of an organization such as cost reduction and productivity improvement (**iv-pi-1**; **iv-pi-4**; **iv-pi-7**).

5.1.2 Driver 2: Efficient value delivery to customers

From the interviews, the second emerged driver for implementing DevOps along with Agile was the ability to deliver the value to the customer as fast as possible with minimum possible cost and without deteriorating the quality of the software being delivered. Agile increases the speed of the upstream processes of the software development such as, identifying business needs and developing the software accordingly (Geurts 2016). On the other hand, the speed of downstream processes such as verifying, validating and delivering the software can be improved with the help of DevOps (Geurts 2016). While Agile helped the business to perceive the past sprint's development during the sprint review, DevOps facilitates better assurance of quality and efficient delivery by bringing the rest of the IT teams closer with Development through cooperation and tooling (Erich et al. 2014b).
"[...]for a bank, ultimately what they want is, faster delivery.. earlier, [with waterfall approach,] to deliver a feature, it was taking a couple of months, [but] they wanted to do it in a much faster time.. I think the DevOps, whole setup is going to help them to do the faster delivery.. because previously, what we used to do is, we have this so many layers.. [for example] the business [team] talks to BA, the BA writes all these documentation, and then it comes to the IT and business knows what is happening only after a couple of months.. if there is any change that has to be done, it is already too late.." (**iv-pi-5**)

5.1.3 Driver 3: Cooperative culture

The next important driver is the cooperative culture, where people really work together with shared goals towards their main objective, delivering value to the customers. With the restricted responsibilities within the siloed organization structure,

people did not have the need to think or have the freedom to work outside their defined roles and responsibilities. Thus the stakeholder perspectives on their work were restricted only to the part of the system development and delivery process, that they were involved with (Iden et al. 2011).
Thanks to the Agile software development process, the wall between the customers and development team was brought down as a consequence of the frequent communication possibilities and smaller iterations. Also the introduction of Product Owner role, who has the responsibility of bringing the business value out of the team's work, has already helped the development roles such as business analyst, developer and tester to align their goals towards delivery of the software. DevOps has enhanced this scenario further by allowing all possible roles within the system development and maintenance to work closely with each other, which has largely enriched the communication among the involved IT stakeholders (**iv-pi-6**). Several studies agree with this and they mentioned this cooperative culture has lead to several positive effects within the organizations such as frequent operational feedback and user feedback, reduced chance of product failure after deployment, improved transparency and collaboration among stakeholders and better work culture (Lwakatare et al. 2016b; Erich et al. 2014b; Erich et al. 2014a; Ravichandran et al. 2016a). Additionally, when followed within the enterprise level, the collaborative nature of the improved work culture reduces re-engineering and thus speeds up the process (Geurts 2016).

"there is more transparency.. everybody is openly telling what is happening to each other.. that is because of the change.. the problems get solved faster.. because you [people] bring impediments to the board faster.. there is more awareness.. everyone is talking and then they know what each other is doing.. these 3 were all hidden before [Agile and DevOps implementation] though work was going on.. but you bring these factors as in, and then you see the biggest advantages in terms of visibility and transformation.." (**iv-pi-6**)

5.1.4 Driver 4: Empowered People

The next identified driver is the upbringing of 'empowered people' within the organizations. As discussed, the less bureaucratic the culture becomes, the more control people get themselves within the organization. In order to make the best use of this potential, teams are supposed to be capable of more than what is described in their job descriptions (iv-pi-3). The environment enables people as a team to work on realizing the goal of the team instead of focusing only on their individual achievements, and it helps them to take responsibility over others' work as well (iv-pi-5). The DevOps environment allows people to share their tasks with each other and so augments their role within the team. Unlike the strict organizational structure, the new DevOps structure encourages people getting experience in as many tasks as possible of those within team's responsibility. Instead of simply executing the assigned work, people are expected to pro-actively participate in pre- and post-execution of their work, and they are demanded to think on a bigger scope without just being constrained to their own application/system/role (**iv-pi-7**).

"I think one of the main reasons is that it[DevOps] will make your team more efficient.. in the sense that, in a way that the team is responsible not just for creating it also for maintaining it.."(**iv-pi-3**)
"besides the team takes the ownership of the application, each individual team member when

they are fully involved in the whole life cycle of the product, so from development until support, they can take ownership themselves.."(**iv-pi-6**)

5.1.5 Driver 5: Focus on continuous improvement

The next emanated driver from the interviews is the increased focus on continuous improvement of organization inclusive of both people and process. Agile promotes incremental software development and delivery (Principle 3) since it allows teams to learn from their past delivery cycle and thus helps them to become better progressively (Principle 12). The collaborative working nature of DevOps teams help people to communicate frequently and so get the feedback on their piece of work faster than ever before, which helps them to correct their mistakes and improve their work quickly (**iv-pi-7**; **iv-pi-5 and iv-pi-3**). The Scrum practices such as retrospective and review meetings, and the emphasis on automation that comes with DevOps thinking can all be considered as results of continuous improvement. The interviewee 1 rightly pointed out the change of focus observed within the Agile and DevOps organizations as following:
"years ago, there was a focus on delivery.. now [ie. after implementation of Agile and DevOps] it is on both delivery and continuous improvement.. so more thinking about how can we do a better job.." (**iv-pi-1**)

5.1.6 Driver 6: Process and stakeholder alignment

The alignment among various stakeholders such as business teams, IT teams and end users is identified as the following driver for Agile and DevOps implementation. Such an alignment requires collaboration and communication among IT teams and customers, and between organizational units themselves. This was identified as the top objective for Agile implementation in various large scale organizations according to the research of Hobbs and Petit (2017). Mostly in agile organizations, project teams are provided with the accountability of delivering software and the management groups such as project manager and product owner have the decision making power. In order to realize the success of the Agile projects, the alignment among the project teams and management groups is significant (Cockburn and Highsmith 2001). In Agile methods, the business team meets the project team regularly atleast at the end of every sprint and this helps them to align with each other (Hobbs and Petit 2017). DevOps allows the Operations team members to be in frequent collaboration with the development team so that they have more awareness about current development and helps them prepare better for the deployment part of the same (**iv-pi-3**; **iv-pi-5**).

5.2 Perspectives and Focus Areas

Next, we are identifying the perspectives and focus areas since they will be helping to form the intended DevOps implementation framework as described in Section 1. The perspectives are identified from the literature as mentioned from Chapter 2, and they are maintained further since they were accepted by the interviewees as appropriate to incorporate all the relevant adoption factors.

The focus ares are the principal sections that require attention within every perspective regarding the Agile and DevOps adoption. These focus areas are patterns identified mainly from the interviews and they are further explained with the corresponding literature studies. The relationship among the involved perspectives can be visualized as given in figure. 5.2.

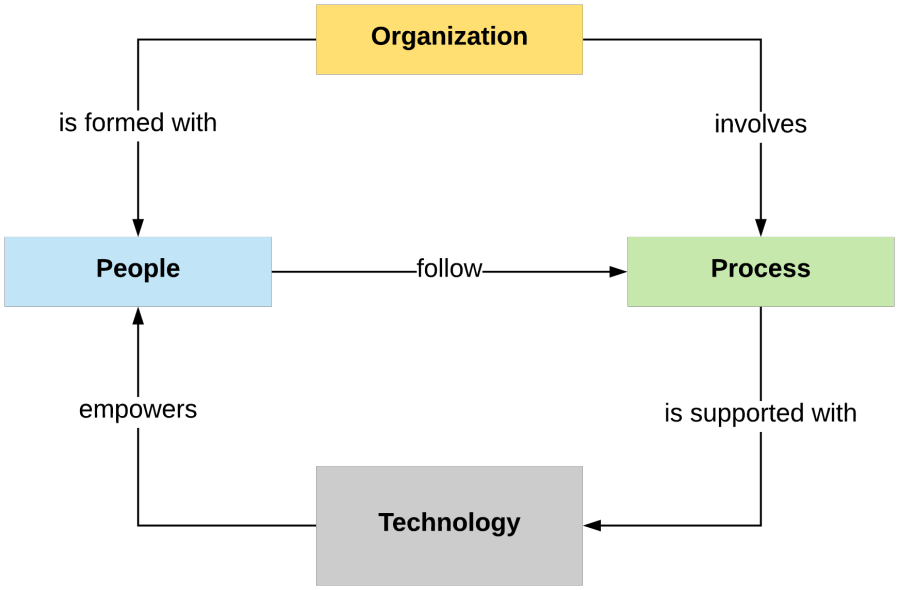

FIGURE 5.2: Relationship among the involved perspectives

5.2.1 Organizational Perspective

The first one is the organizational perspective and it includes all the focus areas that the higher management of a typical non-DevOps enterprise should consider and facilitate in order to constitute the landscape within an organization to foster DevOps mentality. A summary of identified focus areas and the drivers that are related to them can be identified in the following table **??**.

Focus Area 1: (Sub-)organizational Structure

The first focus area within the organizational perspective is about the structure of the organization and the sub-organizations. The large sized financial organizations such as banks have several sub-organizations, which are the smaller units that focus on specific products or services or applications within the bank. These sub-organizations have one or more agile teams. This focus area concerns with the organization of team members within the teams and the organization of teams within the sub-organizational unit.

Concerning the structure of the teams, DevOps leads to teams that bring together experts such as, software development professionals and operations professionals so that, they can share their skills and experiences (Huttermann 2012). The team structure should allow for live and peer-to-peer communication within the team but not via management or tickets (Sharma 2017; Swartout 2014).

TABLE 5.1: Matrix of focus areas related to organizational perspective and the drivers

Focus Area (FA)#	Driver 1: Agility and Customer-centricity	Driver 2: Efficient value delivery to customers	Driver 3: Cooperative culture	Driver 4: Empowered people	Driver 5: Focus on continuous improvement	Driver 6: Process and stakeholder alignment
FA1: (Sub-) organizational structure	X	X	X	X		X
FA2: Agile and DevOps oriented people evaluation			X	X	X	
FA3: Large-scale agile practices	X	X	X			X
FA4: Open and trusted environment			X	X		
FA5: Training and Guidance			X	X	X	
FA6: The leadership commitment	X	X	X	X	X	X

In terms of how the team members are located, following possible capabilities can be determined for the team structure (Swartout 2014, p. 32).

- Everyone should be physically at the same location
- Sending required people to rest of the team
- Using video conferencing

With respect to the way, how Development and Operations teams are integrated, the following four capabilities can be arrived (Huttermann 2012, p. 81).

- **Extend Development to Operations** is the way of applying production-relevant items early and often as part of the development process.
- **Extend Operations to Development** allows collaboration in the sense of streaming information from operations back to development.
- **Embed Development into Operations** requires changing processes across development and operations so that the development activities, goals and team are embedded onto the operations'.
- **Embed Operations into Development** brings operations team into the development team, so that they work closely to provide the best solution possible.

Because this focus area concerns about the construction and composition of the sub-organizations within an enterprise, we believe that this focus area is related to the following drivers:

- **Driver 1:** Agility and Customer-centricity
  Rationale: The flat and networked organizational structure that encourages teamwork by cross-functional linkages, drawing loose boundaries among functional units is identified as one of the main characteristics of agile enterprises (Sherehiy et al. 2007). Although there is no empirical study found that is focusing on the impact of organizational structure on organizational agility or customer-centric behavior, Wendler (2013) has identified organizational structure as one of the concepts related to organizational agility. In addition, both

the research and the collected interview data confirms that the organizational structure is impacted by the DevOps practices (**iv-pi-2**; **iv-pi-3**; **iv-pi-4**; **iv-pi-6**; **iv-pi-7**; Shahin et al. 2017; Nybom et al. 2016).

- **Driver 2**: Efficient value delivery to customers
  Rationale: Faster operation time and quickness in the product/ service delivery are certain capabilities of Agile organizations (Sherehiy et al. 2007). The integration of Development and Operations personnel helps to achieve efficient value delivery to customers whereas agile methods were able to make only the development faster but not the delivery (Virmani 2015).

- **Driver 3:** Cooperative Culture
  Rationale: The DevOps structure which removes or even weakens the silos among Dev, Ops and other roles is found to increase the collaboration. The organizational structures where responsibilities can be mixed or shared are allowing people with different roles to communicate and collaborate more. This fact was supported by both the literature and the interviews (**iv-pi-3**; **iv-pi-4**; **iv-pi-7**; Nybom et al. 2016).

- **Driver 4:** Empowered People
  Rationale: Both the Agile and DevOps stresses to have an organizational environment where the knowledge as well as the control is decentralized; tasks are shared; and high degree of flexibility is observed so that coordination becomes informal and personal. Such an environment is hard to achieve without making appropriate changes in the organizational structure. Thus making appropriate changes in the organizational structure and providing appropriate infrastructure is important to emphasize the culture of empowerment, which helps for the close collaboration among involved stakeholders (Sherehiy et al. 2007; Shahin et al. 2017).

- **Driver 6:** Process and Stakeholder alignment
  Rationale: Agile enterprises are characterized to have close contact with the customers and other relevant stakeholders, while practicing frequent and face-to-face communication to achieve efficient and faster work deliveries (Hameed et al. 2016). Without the right structure within the organization, the alignment among the involved stakeholders and so the involved processes can not be regulated.

Focus Area 2: Agile and DevOps oriented People Evaluation

The next focus area is regarding the people evaluation and performance reviews that are commonly conducted within an organization. Agile and DevOps transformations require people to put certain level of social efforts to make it successful. For example, both DevOps and Agile advocate people interaction, mentoring, team work and knowledge sharing. Thus, the performance evaluation should be considering such non-technical skills besides the technical skills required for the actual job (Conboy et al. 2011). However, it is imperative for the organization to make sure that the method of evaluation is team-based, encourages collaboration over competition among team members and teams, and not conflicting with the behavioral needs of Agile and DevOps (Davis and Daniels 2015). Additionally, the rewarding scheme appreciates the work of the team and encourages them to improve together as a team, but not at the individual level (Walls 2013; Swartout 2014). Furthermore, the

organizations should move on from waterfall-styled annual or semi-annual performance reviews to shorter and frequent feedback cycles like Agile software processes (Davis and Daniels 2015).

360 degree feedback is a method in which anonymous feedback about an employee is gathered from multiple sources of his/her immediate work circle. This can be used by the responsible management to provide a constructive feedback to the corresponding employee so that they will be encouraged to focus on improving them further but not complaining them or making them guilt of their mistakes (Davis and Daniels 2015).

Deduction: The organization's people evaluation system should (1)focus on evaluation of the team for project performance, (2)consider the non-technical skills and other abilities required for the Agile and DevOps way of working, and (3)encourage goals of individuals and teams to be non-conflicting.

The drivers for which this focus area is contributing are given below.

- **Driver 3:** Cooperative Culture
  Rationale: As explained above in the description, cooperative culture can be seeded and grown when it is encouraged by the management. The team can either win or fall, but always together. Chatman and Barsade (1995) highlighted that one of the ways to promote cooperation is changing the reward structure to make the cooperative way of working more alluring (Chatman and Barsade 1995; Ravichandran et al. 2016a; Gill 2002).
- **Driver 4:** Empowered People
  Rationale: The positive feedback on people will lead to reinforcement of their innovative outcomes achieved through gained self-confidence and the motivation to self-development (Quinn and Spreitzer 1997; Huttermann 2012).
- **Driver 5:** Focus on continuous improvement
  Rationale: When the whole team is rewarded for bringing a good product, they will work together to improve the involved processes which would result in better product (Walls 2013). Moreover, providing regular and frequent feedback would result in smaller improvement points, which are comparatively easy to achieve and reflect (Huttermann 2012).

Focus Area 3: Large-scale Agile Practices

According to Williams and Cockburn, "Agile is the best suit for co-located teams of about 50 people or fewer who have easy access to user and business experts and are developing projects that are not life-critical" (Dingsøyr and Moe 2014). However, the increasing popularity of the agile methodology has enticed many larger organizations including banks towards embracing them. Nevertheless, the adoption of Agile at the large scale is not the same as that of the small scale organizations and still the results of such larger implementations were found to be quite successful at several organizations (Hobbs and Petit 2017). Thus, this focus area emphasizes on tailoring the agile practices specific to the organization and following them throughout the organization, not only at the team levels but also at the project and portfolio levels across the enterprise.

There are several scaling frameworks emerging in order to help with the enterprise level Agile transformation. It is important for an organization to choose a suitable framework or internally construct one in order to cope with the scaling of Agile within the complete enterprise. Some of the popular scaling frameworks are described below as per Vaidya (2014).

- **Disciplined Agile Delivery (DAD):** This is a hybrid approach that extends Scrum with several other Agile methods such as Kanban and XP (Vaidya 2014, p. 3). This framework employs four distinct life-cycles and expects the organization to fit into one of these.
- **Large Scale Scrum (LeSS):** This framework helps to apply Scrum to very large, multi-site and offshore product development. The LeSS has two frameworks known as, LeSS Framework-1 (suitable for upto 10 Scrum teams) and LeSS Framework-2 (suitable for more than 10 scrum teams).
- **Scaled Agile Framework (SAFe):** This framework articulates an organization into three levels namely Portfolio, Program and Team levels, and it incorporates Agile and Lean practices at all the three levels.

The other such known methods are Scrum of Scrums, Spotify, Nexus and RAGE. Each of the mentioned methods have a different criteria in terms of team size, supported Agile methods and practices, required technical practices and the organization type (Alqudah and Razali 2016) but all of them attempt to conserve the benefits of Agile but in larger organizational environments. **Thus, it is essential for any organization to do their internal analysis to choose the best suitable framework for their situation or fabricate their own method to scale agile within their enterprise.** The following drivers can be associated with this focus area based on the given rationale.

- **Driver 1:** Agility and Customer-centricity
  Rationale: While the general agile methods like Scrum provide the guidelines for team level practices, the scaled agile practices are necessary to manage the project activities that are beyond the team level. Thus they help to maintain the agility throughout the organization (Vaidya 2014).
- **Driver 2:** Efficient value delivery to customers
  Rationale: When the agile practices are not followed throughout the enterprise, it affects not only those that do not follow but also those that follow due to the inter-dependencies. Thus, the delivery of the agile teams get affected due to their dependency on the work of non-agile teams (**iv-pi-1**).
- **Driver 3:** Cooperative Culture
  Rationale: In the large scale agile development, the inter-team coordination is as important as the collaborative communication that happens within a team (Dingsøyr and Moe 2014). This is difficult to attain without following proper scaled agile practices.
- **Driver 6:** Process and Stakeholder Alignment
  Rationale: Due to the limitations of agile methods that limit to team level practices as mentioned with the above drivers, scaled agile practices are necessary to align the stakeholders and regulate the processes within the larger organization or enterprise(Vaidya 2014).

Focus Area 4: Open and trusted environment

The next focus area insists on having an open and transparent communication between the management levels and the rest of the organization. This focus area emerged from the interviews, where multiple interviewees concurred that several management decisions are not explained enough or the consensus with the team is not expected. The teams following Agile and DevOps approaches are expected to have an open communication to resolve issues early and to have the focus aligned. This is an important characteristic not only for the teams but for the entire organization. Thus the organization should be clearly communicating the goals and objectives of the decisions that involve teams, and keep the metrics visible for everyone so that they can share the responsibility to achieve it together (Davis and Daniels 2015).

Deduction: The organization should formulate and communicate the goals, what they want to achieve and probably the vision, how they want to achieve to all relevant stakeholders and business. The skills and abilities of the people should be trusted, thus their decisions should be respected by listening to them. No transformation will be successful without the presence of trust and respect throughout the organization (Davis and Daniels 2015). Above all, the organization should give a safe environment for people to give their honest opinion and feedback without being afraid of fear or abuse (Ravichandran et al. 2016b; Swartout 2014).

According to the given rationale, the following drivers are influenced by this focus area.

- **Driver 3:** Cooperative Culture
  Rationale: Without trust and respect among the people within the organization, there will be no cooperation irrespective of the presence of tools and infrastructure (Walls 2013).
- **Driver 4:** Empowered People
  Rationale: The successful agile organizational environment supports experimentation, learning and innovation. It reacts positively to the changes in terms of new ideas and technology. Such an environment will not be possible in an organization, where high degree of formality, obedience, hierarchical authority and hierarchical communication is insisted (Sherehiy et al. 2007). The empowered people spring from the organizational culture which promotes openness and teamwork through participation in organizational decision making (Quinn and Spreitzer 1997).

Focus Area 5: Training & Guidance

The following focus area is about handling the human impediments towards change by providing proper training and guidance. Such impediments include lack of knowledge, cultural issues, resistance to change, wrong mindset and lack of collaboration (Gandomani et al. 2014).

To deal with the lack of knowledge impediment, it is crucial that everyone including management, team and customers are all educated about the Agile principles

and the adopted Agile method(s). In order to manage the cultural and collaboration barriers as well as the confrontation, the human facilitators - Agile coaches and Agile champions are found to be of great help since they can influence the people positively and to make the transition successful. Agile coaches can help with the transition even from preparation phase itself; their responsibilities usually include guiding and training management personnels, and they make sure that the transformation is on track. The agile champions are those who facilitate the transformation and motivate others to follow the change process within the organization(Parizi et al. 2014; Gandomani et al. 2014).

Growth mindset is one of the indispensable characteristics of the people within the DevOps organizations. Likewise, encouraging and enabling that characteristic is one of the main responsibilities of the DevOps organizations. People with the growth mindset will believe that with enough time, effort and practice, they can attain the the knowledge and skills that will help them to do their job better (Davis and Daniels 2015). Although bringing up that growth mindset is up to the individuals themselves, the organization can stimulate it by providing the learning environment where, failures are seen as learning opportunities (not for the corresponding individual alone but for the whole team), innovations and new ideas are encouraged and tried out before being dismissed right away and relevant investments are made to up-skill the employees.

Deduction: Without providing the right trainings, support and guidance, organizations can not expect people to adopt to the DevOps way of working and become successful in it. Thus the organizations should analyze and invest on the suitable ways of educating their people on Agile and DevOps as part of the transformation.

The drivers that are given below are impacted with this focus area in accordance with the provided justification.

- **Driver 3:** Cooperative Culture
  Rationale: The use of right experts such as Agile coaches help both teams and management to get understand the strength of working in a united way and how they should act so that the unity will not get affected. Moreover, the trainings help people to have a common understanding on what Agile and DevOps means to their organization and how they should incorporate it in their way of working (**iv-pi-2**; **iv-pi-3**; **iv-pi-6 and iv-pi-7**).
- **Driver 4:** Empowered People
  Rationale: The trainings on Agile and DevOps practices can help people to understand why they undergoing this transition and how will it help them. This awareness helps people to take ownership on themselves to act right during the transition to take advantage of it and to motivate others in their team. The interviewees agreed that the facilitators be of a great help to get adapted to the new organizational culture, understand and utilize the authority that they have at their own work (**iv-pi-7**).
- **Driver 6:** Focus on continuous improvement
  Rationale: As indicated by several interviewees, the Agile and DevOps adoption at larger financial organizations is a time-taking task, which will not happen in a short period of time (**iv-pi-3**; **iv-pi-4**; **iv-pi-6**). However, it is important for the organization itself to allow people to gradually embrace the change by

allowing them (1) to become aware of the transformation, (2) get educated on the relevant topics, (3) accept the new set of practices, (4) put their learning into practice by getting continuous support and practical guidance.

Focus Area 6: The Leadership Commitment

This focus area talks about supporting Agile and DevOps and also practicing Agile at the top level - higher management and executive level of an organization. Rigby et al. pointed out that Agile methodology is suitable for several C-suite activities such as strategy development, resource allocation, cultivating breakthrough innovations and improving organizational collaboration. This helps them to understand how and where Agile works and where not. This will help the senior managers to work on the highly valuable work that can be done only by them. Some examples of such tasks are, creating and adjusting the corporate vision; prioritizing strategic initiatives; simplifying and focusing work; assigning the right people to tasks; increasing cross-functional collaboration; and removing impediments to progress (Rigby et al. 2016).
Horney et al. (2010) lists out the following specific requirements for agile leaders:

- Provide guidance and directions to teams working across time zones, cultures and organizational barriers
- Take more risks by briskly connecting talent and moving information and knowledge around the globe
- Maintain a laser-like focus on employee commitment and engagement across generational, global, cultural and demographic boundaries
- Make collaboration among suppliers, partners, customers, part-time employees and consultants a signature part of organizational culture

The following Agile Model given in figure 5.2 taken from (Horney et al. 2010) provides the five critical drivers and the corresponding skills of leadership agility.

The macro-managers of an ideal Agile environment practice 'leadership and collaboration' rather than 'command and control' management. They set goals and constraints, bring in the collaborative environment where innovation can flourish, and individuals can effectively apply their competence (Cockburn and Highsmith 2001).

In short, the agile leaders should not only support but also practice the agile methods to perform their leadership activities. The agile leaders require to exercise several leadership agility skills for harvesting organizational success. This focus area is expected to have an effect on the following drivers as explained below.

- **Driver 1:** Agility and customer-centricity
  Rationale: Some interviewees raised a concern that their higher management encourages them to follow agile but they are not following it themselves (**iv-pi-3**; **iv-pi-4**; **iv-pi-6**; **iv-pi-7**). Nevertheless, the organizational agility can be achieved only when all the individuals including managers follow agile (Seo and Paz 2015).

THE AGILE MODEL®			LEADERSHIP AGILITY SKILLS
FOCUSED	FAST	Anticipate Change	**VISIONEERING** – creating clarity on the core value proposition of the enterprise engineered into what the workforce does every day to produce desired outcomes for all stakeholders
			SENSING – understanding forces of change that influence stakeholder success and creation of early warming systems of impending change that can impact success
			MONITORING – having effective processes for tracking performance and trends to identify patterns that impact the organization
		Generate Confidence	**CONNECTING** – establishing clear line of sight for all stakeholders with how each can contribute to the enterprise and person success
			ALIGNING – establishing and living the congruence of vision, value, priorities and actions
			ENGAGING – operating with high levels of inclusion and a climate that delivers the discretionary level of effort from all stakeholders
		Initiate Action	**BIAS FOR ACTION** – establishing an execution culture where a sense of urgency around improvement and all stakeholder satisfaction is a basic shared expectation
			DECISION MAKING – creating capability for fast, effective decision-making at all levels
			COLLABORATING – encouraging ideas and gaining insights across organizational boundaries and from multiple stakeholders
	FLEXIBLE	Liberate Thinking	**BIAS FOR INNOVATION** – establishing permission and expectations that innovation is a universal requirement for all stakeholders' participation
			CUSTOMER FOCUS – establishing on-going alignment and understanding of customers to be able to offer business solutions that meet their needs and often identify unrecognized needs
			IDEA DIVERSITY – establishing processes to encourage and secure innovation inputs from all levels and stakeholders in the enterprise
		Evaluate Results	**CREATING EXPECTATIONS** – providing clear and measurable priorities and resources that are aligned for all stakeholders and desired outcomes
			REAL-TIME FEEDBACK – providing timely and accurate feedback on key success measures for all stakeholders
			FACT-BASED MEASUREMENT – using performance metrics grounded in solid information measurement to allow reliable insights and conclusions

FIGURE 5.3: Leadership Agility Skills duplicated from (Horney et al. 2010, p. 32)

- **Driver 2:** Efficient value delivery to customers
  Rationale: The self-organized agile teams can fail to be effective when their leadership is not agile, and when the traditional management practices are applied. The agile leadership should support teams by removing obstacles that prevent them achieving their goals and, the leaders are expected to set the standard and be the role models for others (Parker et al. 2015).

- **Driver 3:** Cooperative culture
  Rationale: The role of management is crucial is transitioning individual work culture into self-organizing teams that value cooperative work culture (**iv-pi-6**; Moe et al. 2010). The effective leaders promote cooperative culture by activities such as spending enough time with the people in the team, declaring and celebrating success with them and balancing the needs of team and individuals (**iv-pi-7**; Parker et al. 2015).

- **Driver 4:** Empowered people
  Rationale: Self-determination, the ability to choose their own work, is a characteristic of empowered people and people can exhibit this ability only when they are not micro-managed (Quinn and Spreitzer 1997).

- **Driver 5:** Focus on continuous improvement
  Rationale: Agile philosophy puts a great emphasis on achieving the customer satisfaction, for which continuous improvement of people and processes are

vital. Several studies identify that the management commitment and support as a principal factor for the success of process improvement within organizations (Zahra et al. 2017; Gatchalian 1997; Kouzari et al. 2015).

- **Driver 6:** Process and stakeholder alignment
  Rationale: As shown in figure.5.3, agile leaders should be a central point of collaboration for all involved stakeholders. A successful agile leader has to manage the stakeholders by connecting with them, gathering insights from them, and giving timely response to them (Horney et al. 2010).

5.2.2 People Perspective

Cockburn and Highsmith (2001) mentions that the most important people factors required for the agile environment are: **amicability, talent, skill** and **communication**. This perspective identifies the most important people characteristics required for the effective working at Agile and DevOps based organizational environments.

A summary of focus areas identified within this perspective and their related drivers can be located in table 5.2.

TABLE 5.2: Matrix of focus areas related to people perspective and the drivers

Focus Area (FA)#	Driver 1: Agility and Customer-centricity	Driver 2: Efficient value delivery to customers	Driver 3: Cooperative culture	Driver 4: Empowered people	Driver 5: Focus on continuous improvement	Driver 6: Process and stakeholder alignment
FA1: Cross-functional skillset	X	X	X	X		X
FA2: Aligned goals and responsibilities	X	X	X	X	X	X
FA3: Communication and collaboration	X	X	X		X	X
FA4: Teamwork	X	X	X	X	X	X

Focus Area 1: Cross-Functional Skill-set

The following focus area is about the characteristics of cross-functional teams and team members' knowledge, skills and abilities for the effective cross-functional collaboration which is required for the DevOps environment of an organization.

DevOps enables cross-functional skill-set in team members since the DevOps teams consist of different skilled professionals such as programmers, QA and operations personnels. These cross-functional teams, which in general, consist of all different skill-set required for an end-to-end task completion regarding a software product or service, allow people within the team to share tasks and so knowledge with each other. This leads to the team situation, where every individual has at least a general understanding and basic knowledge of other's domains besides their own expertise area (Huttermann 2012). Such individuals, who has a deep knowledge in one area but a good understanding of other knowledge areas along with the skills required

for the agile and DevOps environment are known as T-shaped individuals and their representation is shown in Figure.5.4 (Demirkan and Spohrer 2015).

Abidin et al. (2017) identified the following skills as the requirements for the people within the agile teams. In case of DevOps teams, these skills can be considered as the requirements for all roles within the team but not only for the developers. These skills are further described in detail as part of the upcoming focus areas.

- Communication
- Time Management
- Leadership
- Decision Making

Given the corresponding explanation, this focus area is related to the following drivers.

- **Driver 1:** Agility and Customer-centricity
  Rationale: Multi-skilled and flexible personnel, and the cross-functional teams within an organization provide the organizations with the ability to observe and react faster to the changes (Tseng and Lin 2011).
- **Driver 2:** Efficient Value Delivery to Customers
  Rationale: Cross-functional DevOps teams comprise of cross-functional people can allow people to perform tasks outside of their specialty. This leads to the efficient use of resources and also faster delivery of the work items (Brown et al. 2013; Tseng and Lin 2011; Pinto et al. 1993).
- **Driver 3:** Cooperative culture
  Rationale: The ability and interest towards learning other's tasks lead to sharing responsibilities with each other. When the people from different roles use the same language and terms to discuss details, the discussion becomes more valuable and practical. Thus the cross-functional cooperation among team members help to build trust and respect on one another (**iv-pi-4**; **iv-pi-7**).
- **Driver 4:** Empowered People
  Rationale: People empowerment can be at its two ends. From the organization's end, people should be allowed to take their own decisions regarding the tasks related to their roles. At the same time, from the people's end, they should be having the right skills to utilize their empowerment in the right way. This emphasizes on the importance of people having the required soft skills besides their functional skills (**iv-pi-2**).
- **Driver 6:** Process and Stakeholder alignment
  Rationale: Team-working is the central theme of both Agile and DevOps. It was accepted as the mainly required skill by most of the interviewees (**iv-pi-1**; **iv-pi-2**; **iv-pi-4**; **iv-pi-7**). In an environment where people with different specialties work together, it is not enough for a person to be an expert only in his/her own job and without those personal skills required for the teamwork. The knowledge and awareness of other people's work will be of help to speak the same language, get the right understanding of the conversation and thus may help to align roles with each other.

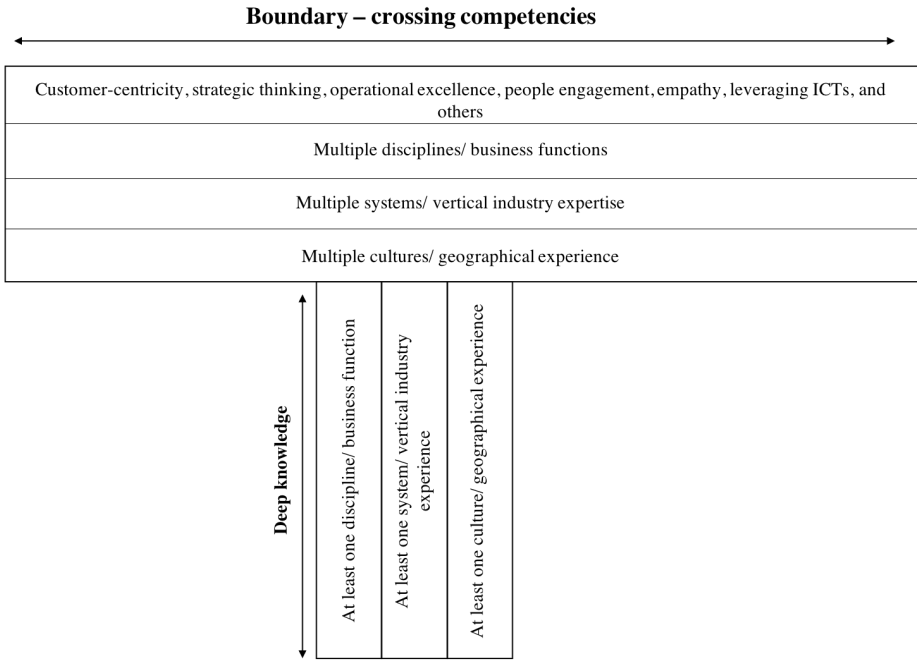

FIGURE 5.4: Infographic representing the skill-set of a T-shaped professional adapted from (Demirkan and Spohrer 2015, p. 13)

Focus Area 2: Aligned Goals and Responsibilities

Huttermann (2012) defines team as a "working group in which members work together closely, cooperatively, and interdependently to achieve a shared group goal" (Huttermann 2012, p. 66). A DevOps organization itself can be considered as a single team working towards the achievement of organizational goals and the definition holds true to every sub-group within the enterprise ie. the team of teams. As mentioned the previous focus area - 'open and trusted environment', a team's goals should be first drawn and then shared to all the relevant stakeholders so that they can work together.

The sub-organizational goal should be aligned to the main goals of the enterprise and in the same way, an (agile) team's goals should be in-line with the corresponding sub-organizational goals. Thus the people goals and their responsibilities should be driven by the shared team goals which are associated to them.

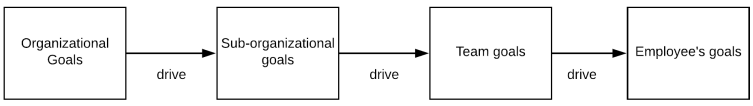

FIGURE 5.5: Infographic emphasizing aligned goals within DevOps Environment

Within a DevOps organization, the team goals should not conflicting with each other but focusing on achieving a common goal that is beneficial to an user group. For example, the following can be an example team goal and its implications:
Goal: Automating the functional test cases of ABC application before end of Q1 2018.

The above team goal can drive the following activities among different stakeholders.

Role: IT Management
Responsibility: Allocate resources (people, time and budget) to let the team work on this automation activity.

Role: Business team/ Product Owner
Responsibility: Reduce the scope of the development activities within the sprint as the team spares time with automation activity.
Role: Team
Responsibilities:
- Develop, test and document the automated functional test cases.
- Use the automated test cases thereafter during the work of the sprint.

The emphasis on aligned goals and responsibilities may contribute to the following drivers as per the given rationale.

- **Driver 1:** Agility and Customer-centricity
  Rationale: The goal-directed coordination is identified as a characteristic of agile enterprises (Sherehiy et al. 2007). Following a goal helps people to maintain the customer-centric behavior (**iv-pi-4**). As an example, picking the task of highest priority instead of the one that an employee is comfortable working with, can be seen as a customer-centric behavior which is instilled by the agile way of working.

- **Driver 2:** Efficient value delivery to customers
  Rationale: The adherence to shared team goals is found to be directly impacting the task outcome of those teams (Pinto et al. 1993). Oppositely, the lack of common understanding of the project's goals is affecting the customer value delivery (Augustine et al. 2005). The interviewees acknowledged the same observation at their work environments (**iv-pi-4**; **iv-pi-7**)

- **Driver 3:** Cooperative culture
  Rationale: According to the researches of Pinto et al. and Ghobadi and D'Ambra, the team member's commitment to the shared project goals and their task orientation strongly influences the cross-functional cooperation of those teams and prevents them embracing competition for both tangible and intangible resources (Pinto et al. 1993; Ghobadi and D'Ambra 2012).

- **Driver 4:** Empowered People
  Rationale: The clear vision and challenge that comes with it drives the empowerment of the people. On the other hand, once the goal is understood, the empowered people will start working for it autonomously but the challenges come with it should provide the opportunity to improve themselves as well as the organization (Quinn and Spreitzer 1997).

- **Driver 5:** Focus on Continuous Improvement
  Rationale: Without an alignment between goals and responsibilities, the continuous improvement is not possible within any organization. DevOps implementation itself can also be seen as a 'continuous improvement' activity, which aims to bring the improvement in people's way of working, organization's results etc.. However, the success of the implementation is not at all possible

without the involvement, support and the relevant action of people within the organization.

- **Driver 6:** Process and stakeholder alignment
  Rationale: It is only when working towards the shared goals, a group of individuals become a team. Additionally, the shared goals help them to align with each other in terms of responsibilities, processes, tools and any other things that are in common.

Focus Area 3: Communication and collaboration

Next focus area accentuates the need for effective communication and intense collaboration among the team members, IT management and business. While communication can be defined as sending and receiving of information, collaboration is the act of working together to achieve a certain goal such as deliver a work product or make a decision (Cockburn and Highsmith 2001).

With the act of communication, people do exchange knowledge, influence each other, recognize each other's work and build a community. By working collaboratively within the community, people build trust and empathy for each other (Davis and Daniels 2015).

By working together within a community, people should be encouraged to lead towards a blameless culture where solutions are focused but not the mistakes. Rewarding the good behaviors and celebration of success should be part of the community's culture. The open and constructive feedback from each other should be appreciated and encouraged. The following particulars regarding working agreements are highlighted by (Huttermann 2012) for fostering collaboration within Agile teams (Huttermann 2012, p. 67).

- An explicit set of agreements about how a team functions provides clarity that prevents confusion and conflict later.
- Teams function together in many ways, and every team does it somewhat differently. You can only assume that everyone shares the same understandings if you have discussed and documented them.
- Working agreements help new members learn how to participate constructively. They serve as the basic list of key dos and don'ts.
- In meetings, explicit working agreements help members stay accountable, because if they violate an agreement, any other member can point out what they agreed to.

The following are such workplace agreement examples taken from (Huttermann 2012).

- *Knowledge transfer:* Developers pair with members of operations team and vice versa. Pair only once per day.
- *Time:* When team meetings are set, make an effort to attend, be on time, come prepared, help the team stay on task.
- *Wise use of meeting time:* Stick to one conversation at a time in meetings.

Deduction: Like above, the teams can come up with the workplace agreements that is suitable for their team situation. The agreements can be updated whenever the team members feel that it is necessary. Moving forward, any team member may ask for a test for an agreement at anytime. Others may clarify the question that was raised, and all indicate their level of agreement.

- **Driver 1:** Agility and Customer-centricity
  Rationale: Internal cooperation of people within the organization as well as the close collaboration with other stakeholders such as customers and suppliers are identified as the key characteristics of agile based organizations (Sherehiy et al. 2007). Working closely with customers and having frequent customer-interaction makes the agile way of working more customer-centric. This characteristic is augmented by DevOps way of working, since customers are associated not only with the development but also delivery and maintenance activities of the team.

- **Driver 2:** Efficient value delivery to customers
  Rationale: While quick development and timely delivery of products are identified as the capabilities of agile organizations, DevOps improves it by enabling efficient delivery of products to customers by reducing the communication gap between the Development and operations and other stakeholder, and by promoting process automation (Sherehiy et al. 2007; Virmani 2015).

- **Driver 3:** Cooperative culture
  Rationale: Regular communication along with task orientation and interpersonal relationships are found to be the keys for cross-functional cooperation (Ghobadi and D'Ambra 2012). The improved communication and so enhanced collaboration among different roles such as developers, testers, operations personnels, product owner and management was found to be a great advantage of DevOps practices at agile organizations (**iv-pi-3**; **iv-pi-4**; **iv-pi-5**; Riungu-Kalliosaari et al. 2016).

- **Driver 5:** Focus on Continuous Improvement
  Rationale: The continuous improvement can be initiated by adopting blamelessness where people analyze the causes of issues and try to improve them without simply pointing fingers towards each other. Furthermore, the problems identified as such should be communicated and dealt appropriately in a timely fashion. This way continuous improvement can become an integral part of the organizational way of working (Bessant et al. 2001).

- **Driver 6:** Process and Stakeholder Alignment
  Rationale: Clear and open communication among teams and other stakeholders help to avoid unforeseen circumstances and thus reduces friction (**iv-pi-7**; **iv-pi-4**). In essence, transparent and timely communication help people to align their expectations, adjust their planning so that the seamless cooperation among them will be in place.

Focus Area 4: Teamwork

The next focus area draws attention to the teamwork aspect of people working in the agile and DevOps organizations. Team work boosts not only the performance of the team but also the individual performance, which in turn helps for the performance

of the organization.

The Dickinson and McIntyre's teamwork model Figure. (5.6) taken from Moe et al. is used as the reference model to explain the important concepts involved in the team work and also to show the relationship among them.

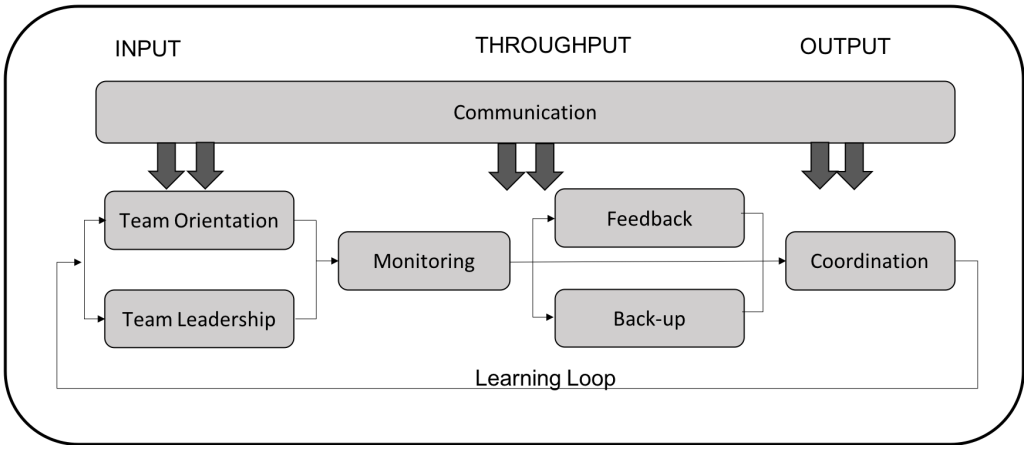

FIGURE 5.6: Dickinson and McIntyre's teamwork model adapted from (Moe et al. 2010, p. 482)

The main components of the model are explained below according to the explanation taken from Moe et al. (2010).

Team Orientation. It refers to the behavior of the team members towards each other and their actions towards the team's collaboration. Following are a few examples (Moe et al. 2010).

- Giving more importance to the team goals
- Actively participating in the relevant aspects of the team

Team Leadership. This involves the leadership activities such as providing direction and support to other team members, bringing structure to the team activities etc. In the self-managing agile teams, this leadership has to be shared by multiple or all team members. Some of such team leadership activities are,

- helping other team members to understand their task within the current developmental activity
- listening to the concerns of the other team members

Monitoring. It is about observing the team activities, the performance of other team members and recognizing them whenever required. Thus it helps the team members to provide the right feedback to other team members or to offer support to others by being a back-up to their activities.

Feedback. This is regarding seeking and receiving feeback from, and providing feedback to the rest of the team.

- Team members should be able to accept the negative feedback as well as the positive information about them.
- The constructive feedback should be practiced focusing on the activities without labeling the team members (Davis and Daniels 2015)

Backup. This is about being available to provide support for other team member's tasks. This also implies that the team members are aware about each other's work and their knowledge is shared so that they can assist one another whenever needed. The previously discussed 'cross-functional skillset' of the team members helps for the backup ability of the team members.

- Filling-in for other team members whenever needed
- Sharing tasks and knowledge with each other

Coordination. It refers to the timely execution and cohesive performance of team members' activities. In order to be coordinated, the team members need to communicate, influence the work of other team members and so being committed to the shared goals as mentioned in the focus area 'Aligned Goals and Responsibilities', help them to coordinate better (Davis and Daniels 2015).

Communication. It is the exchange of information between two or more team members. This aspect is involved in all other aspects of the teamwork as shown in the Figure. 5.6 and the previously discussed 'Communication and Collaboration' focus area can be accessed for more information on this.

The previous focus areas discussed in detail about the cross-functional skillset (Focus area 1), sharing goals and responsibilities (Focus area 2) and communication and collaboration (Focus area 3)of the team. The model from Figure.5.6 connects all those focus areas and positions them appropriately within the teamwork environment. Moe et al. had used this model for performing a case study in an Agile environment and thus this model had already been found to be suitable for that.

Deduction: The presented teamwork model outlines the important components of teamwork and it can be encouraged to be followed by the people to become aware of the expectations within such an environment that demands intense communication, collaboration and cooperation. This is only a guiding model and the teams can develop their own model or update the given one and follow it to make sure every personnel has the same level of understanding of the team work.

Agile approaches all emphasize the need for distributed leadership through the creation of empowered teams, where each member takes personal responsibility for the achievement of the team.

- **Driver 1:** Agility and Customer-centricity
  Rationale: One of the success factors for Agile is self-organizing teams and the strength of self-organizing teams is the 'teamwork' (Moe et al. 2010).
- **Driver 2:** Efficient value delivery to customers
  Rationale: Several literatures as well as the interviewees, all agreed that the teamwork positively contributes to the efficient value delivery to customers (**iv-pi-4**; **iv-pi-5**; **iv-pi-6**; Lindsjørn et al. 2016; Ravichandran et al. 2016a; Moe et al. 2010). While agile encourages teamwork within development teams, DevOps extends it to operations and other traditional functional divisions (Lwakatare et al. 2016a).
- **Driver 3:** Cooperative culture
  Rationale: The cooperative work environment is a result of team work since the it involves the components such as communication, coordination, cohesion

and mutual support, which all contribute to the cooperative culture (Moe et al. 2010; Lindsjørn et al. 2016).

- **Driver 4:** Empowered people
  Rationale: The self-organizing teams consist of empowered people, who has the strong sense of responsibility towards their own work assignments as well as the their team's work. They commit themselves to the team tasks and perform it by coordinating with other people within as well as outside the team. Both the interviewees as well as scientific literature agree that the cooperative work of empowered people as a necessity for the success of modern organizations (**iv-pi-4**; **iv-pi-6**; **iv-pi-7**; Gatchalian 1997; Lindsjørn et al. 2016).
- **Driver 5:** Focus on Continuous Improvement
  Rationale: When continuous improvement is seen as a regular outcome of an endless process whereby steady innovations emerge, it is achievable only when people at all levels within the organization support and strive towards it (**iv-pi-2**; Bessant et al. 2001).
- **Driver 6:** Process and Stakeholder Alignment
  Rationale: The teamwork is the result of people aligning with each other within a team. Within an organization, people alignment is not only required within the team but also outside the team whenever dependencies are in place. Certain interviewees mentioned that they try to keep this interdependencies as minimal as possible (**iv-pi-6**). Nevertheless, considering a complete enterprise as a team of teams, the alignment among the teams is crucial for the success of the enterprise and it is not possible without the teamwork.

5.2.3 Process Perspective

This perspective includes the important process areas that the agile organizations need to consider within the context of DevOps. There are three such focus areas derived for the process perspective namely, Change management, Knowledge Management and Continuous Process Improvement. The association between these focus areas and the identified drivers are summarized at the following table 5.3.

TABLE 5.3: Matrix of focus areas related to process perspective and the drivers

Focus Area (FA)#	Driver 1: Agility and Customer-centricity	Driver 2: Efficient value delivery to customers	Driver 3: Cooperative culture	Driver 4: Empowered people	Driver 5: Focus on continuous improvement	Driver 6: Process and stakeholder alignment
FA1: Change and operations management	X	X				X
FA2: Knowledge management		X	X		X	X
FA3: Continuous process improvement			X	X	X	X

Focus Area 1: Change and operations management

This focus area insists on developing a change and operations management framework and integrating it with the project management method. DevOps practices intend to reduce the time between the code commits of a change in the development system and placing the change in the production system (Riungu-Kalliosaari et al. 2016). The changes can be due to bugs, enhancements or change requests of the system (Huttermann 2012). Traditional change management processes can delay the release for multiple days although the features could be developed and in ready-for-release state (Chen 2015). However, this time period can be reduced and the probability of change acceptance can be improved by involving operations groups in the Change Advisory Board and by coordinating with operations maintenance (Phifer 2011). Following are some example activities in that regard.

- Change review of prospective solution designs by infrastructure engineering and operations groups to make sure that the current operations will not be negatively impacted.
- ITIL organizations include change in the 'Forward Schedule of Change' to ensure coordination with other planned infrastructure change and maintenance activities

Mohamed (2016) has identified the following criteria that should be assessed regarding the change management of any DevOps organization. As per the author, the following criteria have to assessed to identify the corresponding CSF's and KPI's, and the corresponding actions need to be taken accordingly.

- Change management strategy
- Change management control board formulation
- Change automation
- Review process for each change
- Implementation of feedback loops
- Data migration strategy

Franklin (2014) has given the following general guidelines to develop the custom change management framework for an organization.

- Make all the stakeholders aware what change management is and why is it important
- Develop an change management framework which includes all activities (end-to-end)and techniques, and integrate it with the project management method
- Train Project Managers, Business Analysts and other relevant stakeholders on the change management framework
- Start using the framework and improve it based on the lessons learned every time.

Deduction: The organization should develop the change management framework that is suitable for them based on the input from all relevant stakeholders including teams, and follow it. The above given guidelines and criteria can be considered regarding the development of such a framework.

This focus area can be related to the following drivers according to the given justifications.

- **Driver 1:** Agility and customer-centricity
  Rationale: From the interviews, it was understood that every team or department requires a different change management strategy due to several reasons such as frequency of changes requested, nature of the application, customer etc (**iv-pi-1**; **iv-pi-3**; **iv-pi-6**). When an organization wants to be highly customer-centric, it is very important to satisfy their needs in a deliberate manner. Hence, an appropriate change management framework is important so that all the involved stakeholders are aware about the unified way of handling the changes.
- **Driver 2:** Efficient value delivery to customers
  Rationale: When there is a proper change management process in place, the involved stakeholders such as management or customers may try to influence the sprint work by introducing last-minute changes. This may impact both the quality of the product and the planned work of the team. As a consequence, the project delivery may get affected (**iv-pi-3**; **iv-pi-6**). Thus, it is necessary to have the proper change management process, which is agreed and followed by all involved stakeholders which in turn is important for the timely delivery of quality software products.
- **Driver 6:** Process and Stakeholder Alignment
  Rationale: The alignment among all involved stakeholders such as team, customers and IT management is required for establishing a change management framework as mentioned by (Franklin 2014). Moreover, the developed change management framework has to be in-line with the other processes such as requirements refinement and, development and testing of requirements.

Focus Area 2: Knowledge Management

Thanks to Agile and also DevOps, the functional groups of people are disseminated and restructured into cross-functional DevOps teams, which are formed around value streams. This brings in a clear need for effective knowledge management processes and activities so that continuous learning and coordination can happen within the enterprise. So, this focus area elaborates on the subject of knowledge management and highlights the relevant notes from the different literature.
Knowledge management is the process of 'gathering, analyzing, storing and especially sharing knowledge and information within the organization'(Phifer 2011). Sharing of tools can also be included in the context of knowledge management in DevOps organizations due to the central role of automation in it. The knowledge management has a prime position in the DevOps culture, where trust and openness plays an important role as discussed in the organizational and people perspectives earlier(Phifer 2011). According to McGinnis and Huang, the knowledge within an organization can be at three levels namely, individual, group and organization. Considering that, the knowledge sharing can be happening in three ways as below.

- **Knowledge sharing between individuals** : Also called as socialization where two or more individuals share tacit knowledge i.e. the kind of knowledge which is difficult to be written down (McGinnis and Huang 2007).

- **Knowledge sharing between groups or teams** : The knowledge being transferred among between groups of individuals and it includes both tacit and explicit knowledge (the knowledge that can be documented (McGinnis and Huang 2007))
- **Knowledge sharing between organizations** : The knowledge being transferred between different organizations by utilizing their resources

The team level knowledge sharing may happen in two ways as given below (iv-pi-2, iv-pi-3, iv-pi-4 and iv-pi-6).

Intra-team knowledge sharing. This is about sharing knowledge among people within the team. This helps the people within the team to acquire knowledge about the technology, processes (requirement analysis, performance testing etc.) and techniques that the other team members are working with.
Examples: Team retrospective meetings, Pair programming, personal interactions

Inter-team knowledge sharing. This is regarding the sharing of knowledge among the communities of people who share their expertise in terms of the technology or the role.
Examples: Knowledge sharing networks such as Community of practice (CoP), seminars and workshops

One of the interviewees mentioned about having a separate automation team within their organization, which helps across sub-organizational units (a.k.a departments) to bring-in the automatic deployment pipeline (**iv-pi-6**). They can be considered as a knowledge sharing network since they are collecting the new automation related solutions and tools provided by one team and help to spread it to other teams. Moreover, being a single group of automation experts, they can provide advise and help regarding any automation related tasks required within the teams. Thus, the expert knowledge is shared throughout the organization and this can be identified as a perfect utilization of team level knowledge sharing.

As per the research of Wickramasinghe and Widyaratne (2012), the knowledge sharing in project team work environment is positively influenced by the following factors:

- Interpersonal trust
- Perceived rewards towards knowledge sharing
- Work-group communication mechanisms such as brainstorming and collective problem solving
- Personal interactions such as wiki, emails and chats

Deduction: The organization and the people need to identify the suitable knowledge management processes that work for them and support them with relevant tools and infrastructure. People should be aware of the advantages and importance of knowledge management practices and so encouraged to share knowledge with each other.

This focus area can be associated with the following drivers as per the given rationale.

- **Driver 2:** Efficient value delivery to customers
  Rationale: While individual knowledge sharing help to build a team culture, inter-team knowledge sharing help to avoid re-engineering. Overall, knowledge sharing helps for efficient work delivery of organizations. Several other studies appreciate the fact that DevOps provides an opportunity to share knowledge and tools among different functional roles within an organization and that being an important reason behind efficient value delivery as a result (Davis and Daniels 2015; Swartout 2014; Huttermann 2012; Walls 2013).
- **Driver 3:** Cooperative culture
  Rationale: Sharing of tools, information and knowledge is a part of the DevOps culture (França et al. 2016). This sharing makes the organizational culture into cooperative but not competitive. In addition, following formal knowledge management strategies may help people to consider knowledge sharing as one of their responsibilities (Ghobadi and D'Ambra 2012).
- **Driver 5:** Focus on continuous improvement
  Rationale: Only when the existing knowledge is shared and applied, it makes a way to revise and update that knowledge (**iv-pi-7**; McGinnis and Huang 2007). In this way, the knowledge management helps to improve both people and process dimensions within an organization.
- **Driver 6:** Process and stakeholder alignment
  Rationale: According to McGinnis and Huang (2007), people within an organization play an important role in the knowledge management. Without proper support, involvement and practice of people, knowledge management can not happen successfully within an organization. Moreover, Agile methods incorporate several processes that involve knowledge management activities. However, they can all be fruitful only when practiced them correctly.

Focus Area 3: Continuous Process Improvement

Process improvement is a primary subject in both Agile and DevOps phenomena. Cockburn and Highsmith (2001) describes that, 'agile processes are designed to capitalize on each individual and each team's unique strengths: One-size-fits-one — every process must be selected, tailored, and adapted to the individuals on a particular project team'(Cockburn and Highsmith 2001, p.132). Smeds et al. (2015) mentions that the DevOps helps to improve the planning and development processes with the help of data from operations. Hence, this focus area aims to highlight the importance of process improvement activities in Agile and DevOps organizations.

By design, Agile methods offer several possibilities that enable continuous improvement. For example, Scrum events such as daily scrum, retrospective and review meetings all intend to improve the processes by finding out the impediments as early as possible. However, the experienced agile practitioners should be allowed to customize the agile practices in order to make sure that it suits for their environmental needs. This is because, not only the project requirements but also environment can keep changing due to several reasons such as people and culture. Thus, the agile processes need to adjusted, tailored and improved to suit the current situation (**iv-pi-7**; **iv-pi-6**; Rigby et al. 2016; Cockburn and Highsmith 2001).

The close communication between the development and operations' roles in the DevOps environment allows to utilize several monitoring data such as infrastructure monitoring and user behavior monitoring data during the early phases of planning and development processes. These feedback loops facilitate the experimentation and the continuous improvement within the engineering process (Smeds et al. 2015).

The metrics play an important role in measuring the process outcome. When required, the organizations can perform 'Root Cause Analysis' (RCA) to identify the issues that are happening at any stage of the processes. This helps to identify the root-cause of the issue and thus helps to figure out the corrective actions by the corresponding stakeholders. Following are two possible ways to perform this analysis according to Davis and Daniels (2015):

- **5-Why's analysis:** Asking 'why' until the root causes are identified.
- **Fishbone (a.k.a. Ishikawa) diagram:** Systematic identification of causes and grouping them into major categories

Deduction: The Agile and Devops adoption by large complex organizations require experimentation and adaptation of the methods and processes to the organization's structure, culture, product/service strategy, human resource management policies, customer interfaces, project roles and governance structures, including program and project portfolio management (Hobbs and Petit 2017).

This focus area can be related to the following drivers as per the given rationale.

- **Driver 3:** Cooperative culture
  Rationale: As mentioned with previous focus areas within people perspective, any group goal can be achieved with the support and participation of all involved stakeholders in and outside the group. Continuous process improvement is an ongoing process but it has to be considered as one of the goals of all the involved people. Thus, maintaining the proper communication and cooperation of people is important to keep improving the process and thus maintaining the high quality. Several studies as well as the interviewees mentioned about the importance of group culture and people behavior towards continuous process improvement (**iv-pi-1**; **iv-pi-6**; **iv-pi-7**; Kouzari et al. 2015; Zahra et al. 2017).
- **Driver 4:** Empowered People
  Rationale: The improvement within an organization starts with one's own work. When the employees are given the opportunity to plan and perform their own work, they will get to use their ability to improve the processes involved (Gatchalian 1997). One of the interviewees mentioned that process improvement is something to be initiated by the people who use those processes themselves (**iv-pi-6**). A good example for this is the automation of processes. Until teams report of the processes that need to be automated and their justifications for the automation need, the management might not be aware of it. However, the team should be encouraged to identify the improvements required and communicate them appropriately.
- **Driver 5:** Focus on continuous improvement
  Rationale: Focus on continuous process improvement is necessary to keep the software process quality high, which in turn can help to produce better quality

software (Almomani et al. 2018). Moreover, the tailored processes give more benefits than the standard ones (Kouzari et al. 2015).

- **Driver 6:** Process and stakeholder alignment
  Rationale: As indicated previously, continuous process improvement is not possible without the support of people. When people are involved, they certainly have to align themselves to keep achieving a goal. A recent systematic literature review study identifies that the involvement and support of stakeholders including team and management influences the success of software process improvement (Zahra et al. 2017).

5.2.4 Technology Perspective

This perspective identifies and describes the focus areas which require attention from the technological standpoint within the DevOps implementation. The two main focus areas identified as such are (1) Tooling and automation; (2) Continuous software engineering practices.

TABLE 5.4: Matrix of focus areas related to technology perspective and the drivers

Focus Area (FA)#	Driver 1: Agility and Customer-centricity	Driver 2: Efficient value delivery to customers	Driver 3: Cooperative culture	Driver 4: Empowered people	Driver 5: Focus on continuous improvement	Driver 6: Process and stakeholder alignment
FA1: Automation and tooling	X	X			X	X
FA2: Continuous Software Engineering Practices	X	X	X	X	X	X

Focus Area 1: Automation and Tooling

Automation is found to be the technological enabler of DevOps. Riungu-Kalliosaari et al. highlights that one of the core DevOps principles is 'the use of agile principles and automation to configure and manage deployment environments', whereas (Huttermann 2012) mentions that 'automation is an essential backbone of DevOps' (Riungu-Kalliosaari et al. 2016; Huttermann 2012). Since several other literature as well as the interviewees agreed with the importance of automation in DevOps, this focus area accentuates the same with more details.

Although the organizational and people factors are the most influential in the Agile and DevOps adoption, usage of tools and the automation provide a significant contribution to the process factor since they provide several benefits such as, (1) improve the process quality by reducing the manual errors; (2) provide the traceable solutions by giving the ability to track all the executed steps; (3) reduce time on performing the most repetitive and most time-consuming activities; and (4) produce consistent and repeatable results and make the process transparentHuttermann.
Huttermann advices that the automation activities should be driven by technical

considerations and they must result in concrete benefits. Thus the following pitfalls need to be considered before deciding on the level of automation (Huttermann 2012).

Law of marginal costs. It has to be considered that the automation system development and maintenance costs time and money.

Verb/Noun mistake. Testing is a process ie. verb but the test cases are artifacts i.e.noun. This has to be understood to avoid unnecessary work.

Paradox of automation. Huttermann advises to be aware of the paradox of automation which says that human involvement becomes less but critical when the level of automation gets higher in a system. This is because, when the automated system is unavailable due to any error, it may take a lot of time to identify and fix the error. Thus strong skills are required to maintain the automated system.

Irony of automation. High automation may result in amateur human operator who cannot fix the system when it fails.

According to the above, the organization should perform efficient automation which makes humans more crucial but not novice, while considering the involved cost, time and other possible pitfalls as discussed.

In accordance with the given explanations, this focus area can be related to the following drivers.

- **Driver 1:** Agility and customer-centricity
  Rationale: Automation of key process segments help to reduce the time-to-market, shortens the response time for change requests and customer feedback which all lead to improved customer satisfaction (Babar et al. 2017). The cross-functional collaboration within DevOps teams help to unify the process automation and the usage of tools (**iv-pi-4**).

- **Driver 2:** Efficient value delivery to customers
  Rationale: Reduction of cycle time (time taken between the initiation and the delivery of a work item) is identified one of the major improvements identified by DevOps implementations and the role of automation is significant in it (Erich et al. 2014b; Lwakatare et al. 2015; Ravichandran et al. 2016a). Several authors agreed that this ability to deliver faster leads to more frequent releases which together lead to efficient delivery of value to the customers (Erich et al. 2017; Sharma 2017; Ravichandran et al. 2016a).

- **Driver 5:** Focus on continuous improvement
  Rationale: Automation enables identifying the most-error-prone and the most-time-taking process areas and automate them in order to gain all the advantages in terms of time, money and value (**iv-pi-4**; **iv-pi-6**).

- **Driver 6:** Process and stakeholder alignment
  Rationale: One of the interviewees mentioned that the people including team members and management need to initiate the automation within their organization. As mentioned with the previous focus area - 'Aligned goals and responsibilities', people need to make automation as their team goal and take responsibilities in order to accomplish this. This is not possible without the alignment of all involved stakeholders and the corresponding processes.

Focus Area 2: Continuous Software Engineering Practices

Although the relationship is not very clearly identified, many studies include continuous software engineering (SE) practices within the context of DevOps(Fitzgerald and Stol 2014; Sharma 2017; Huttermann 2012) and its magnitude was acknowledged by the interviewees as well (**iv-pi-3**; **iv-pi-4**; **iv-pi-6**; **iv-pi-7**). Thus the next focus area aims its attention towards the continuous software engineering practices within the agile and DevOps organizations.

As discussed in Chapter 2, there are several continuous software engineering (SE) practices detailed in the literature such as continuous planning, continuous security and continuous compliance. However, continuous integration and continuous delivery are mentioned as the two core practices that are primarily required for agile based DevOps organizations while others are mentioned as supporting capabilities (Sharma 2017). The figure.5.7 shows the relationship between some popular continuous SE practices and their span across the software environment. Continuous delivery means that every change in the code is proven to be deployment-ready at any time whereas, continuous deployment means that the code will be deployed in the production. According to this distinction, Continuous delivery is mentioned as a mandatory capability whereas, continuous deployment is optional Sharma, p.19.

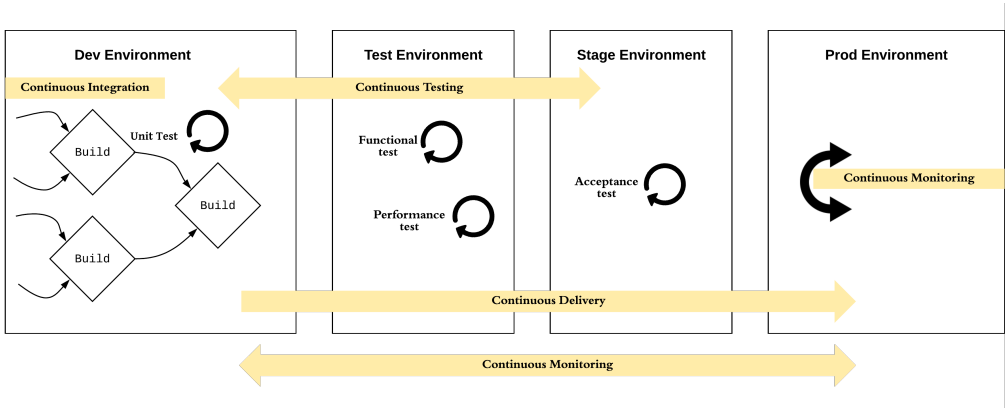

FIGURE 5.7: Infographic on Continuous Software Engineering (SE) practices adapted from (Sharma 2017, p.17)

The given matrix in figure.5.5 from (Farid 2017) highlights how different continuous SE practices employed in the DevOps environment help to eliminate the given lean software development wastes. The mentioned wastes are briefly described below.

- **Delays:** The waiting time that delays work of different roles. Main causes were identified as lack of communication, understanding and trust among stakeholders.
- **Extra features:** Additional features added by team or customers which do not bring expected business value.
- **Extra processing:** Unnecessary knowledge transfer and documentation of applications between departments for example, from Developers to Operations, which can cause time delay and might lead to poor knowledge lead and knowledge rediscovery.

- **Motion:** Lack of information transparency and indirect communication.
- **Incomplete work:** Poor requirements analysis and late discovery of errors may lead to partial completion of features or service.
- **Task switching:** Switching between tasks due to issues such as insufficient knowledge and incomplete planning.
- **Defects:** Poor testing and incomplete requirements that lead to defects.

TABLE 5.5: Matrix of Lean software development wastes and DevOps practices (Farid 2017, p.270)

Lean Waste vs DevOps Practice	Continuous Planning	Continuous Feedback	Continuous Delivery	Continuous Integration	Continuous Testing	Continuous Monitoring
Delays	X	X			X	
Extra features	X	X	X		X	X
Extra-processing (Re-learning)	X					
Motion	X	X		X		
Incomplete work	X		X			
Task switching	X	X				
Defects	X				X	X

Deduction: The organization should identify and implement the mandatory and the most relevant continuous SE practices so that the mentioned wastes can be reduced or eliminated, thus efficient work results can be achieved.

This focus area can be related to the following drivers as per the stated rationale.

- **Driver 1:** Agility and customer-centricity
  Rationale: Continuous SE practices enable the frequent delivery of features to customers and thus entitles faster feedback from the customers or users. This faster feedback cycle impacts the rest of the development and helps them to build the right product according to the wishes of the customer. This flexibility is supported by Agile development process and thus leads to better customer satisfaction (Chen 2015).
- **Driver 2:** Efficient value delivery to customer
  Rationale: The continuous SE practices enable shorter feedback loop in every stage of the continuous delivery and deployment process, which in turn leads to reduced time between the identification and solving of an issue (Shahin et al. 2017; Sharma 2017; Chen 2015).
- **Driver 3:** Cooperative culture
  Rationale: Due to the necessity, DevOps practices such as continuous delivery demands more frequent collaboration among different roles as well as require them to share certain responsibilities. This leads to the cooperative work culture, where people work together on a task where everyone involved is contributing (Shahin et al. 2017).
- **Driver 4:** Empowered People
  Rationale: Sharing of responsibilities as well as continuous SE practices tend to reduce the manual work done by testers and operations personnel. This situation helps them to utilize their time to get awareness about the tasks and responsibilities of other roles such as developers, bring them to be in close

collaboration in with other stakeholders. Besides that, the DevOps way of working enable them to voice their opinion about the design and development decisions such as deployment pipeline (Shahin et al. 2017).

- **Driver 5:** Focus on continuous improvement
  Rationale: The continuous SE practices such as continuous monitoring will help to identify the behavior of the application or system and allow analysis of several technical metrics, so that the system health can be continuous monitored and improved (Sharma 2017).
- **Driver 6:** Process and stakeholder alignment
  Rationale: Each of the continuous SE practice is specific to certain roles (but not titles). For example, Continuous integration is a practice meant for developers, whereas continuous deployment is an operations practice and continuous release is a business practice (Stahl et al. 2017). Furthermore, these practices can have logical dependencies. For example, continuous deployment is the automatic deployment of software to some environment but not necessarily to customers; nevertheless, continuous delivery is the automatic releasing of software to customers (Fitzgerald and Stol 2014). Thus the alignment among the corresponding processes as well as the involved stakeholders is certainly important for the seamless implementation of continuous SE practices.

5.3 Development of the DevOps implementation framework

5.3.1 Artifact 2: The generic DevOps implementation framework

The generic competence model given in figure 5.8 has been built by arranging the identified perspectives and their corresponding focus areas. This model has been built with the aim of educating the people within the financial organizations that undergo Agile and DevOps implementation. The model can also serve as a reference framework highlighting the primary subjects of interest in terms of focus areas within organizational, people, process and technology perspectives.

5.3.2 Artifact 3: Relationship between the identified drivers and the competence model

In this subsection, according to the rationale explained within every focus area, the proposed DevOps implementation framework has been adjusted to highlight the relevant focus areas that require primary focus per driver.

These developed variations do not really mean that the highlighted are the only required focus areas that are required for the achieving the corresponding drivers. Rather, the highlighted focus areas are found to be impacting the driver or related to the driver according to the data collected from the interviews andor from the literature studies.

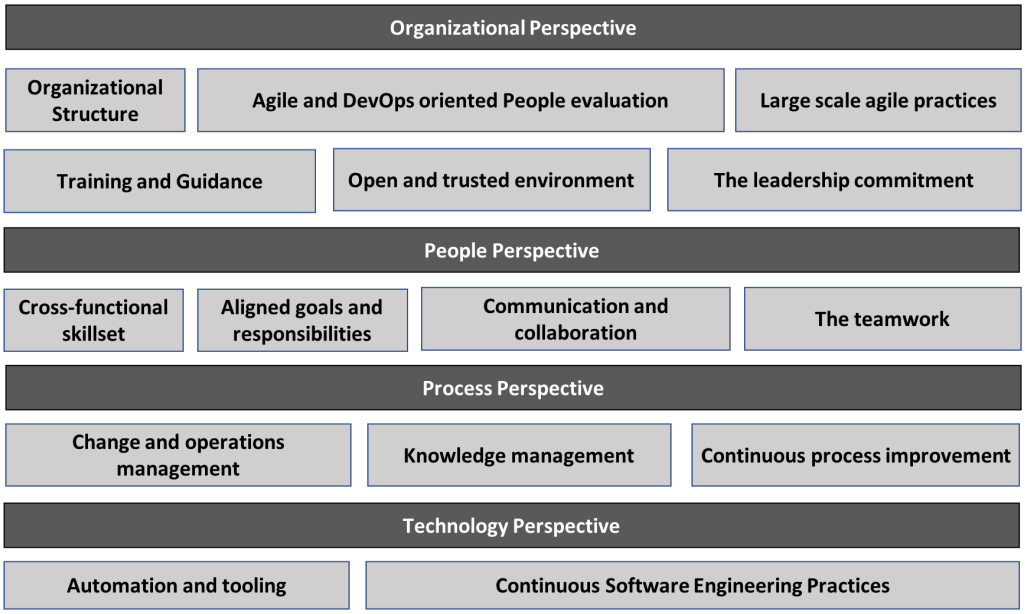

FIGURE 5.8: Proposed DevOps implementation framework for Agile based large financial organizations

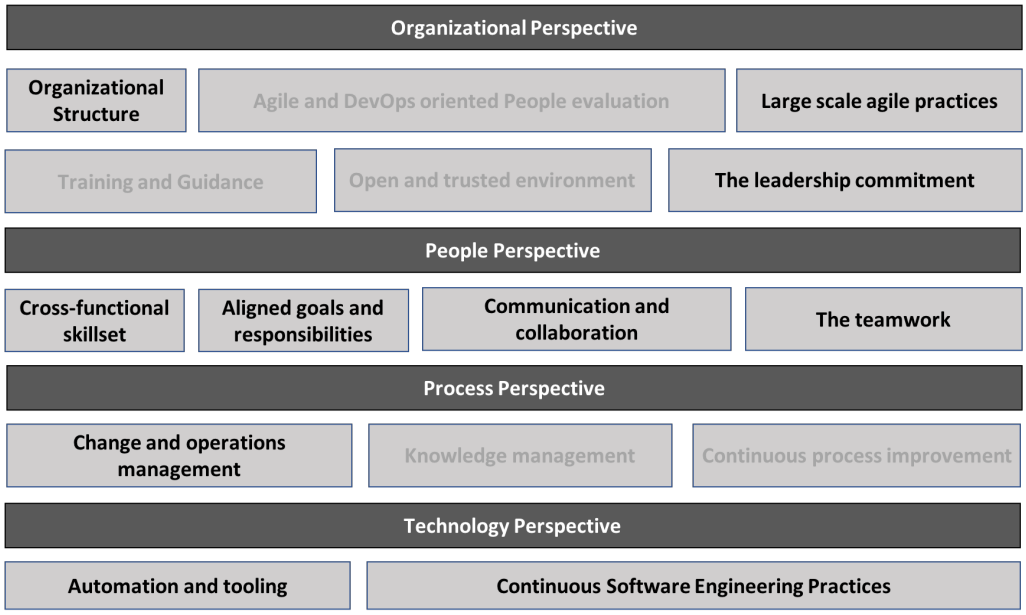

FIGURE 5.9: DevOps implementation framework variation highlighting the focus areas specific for Driver 1 : Agility and customer-centricity

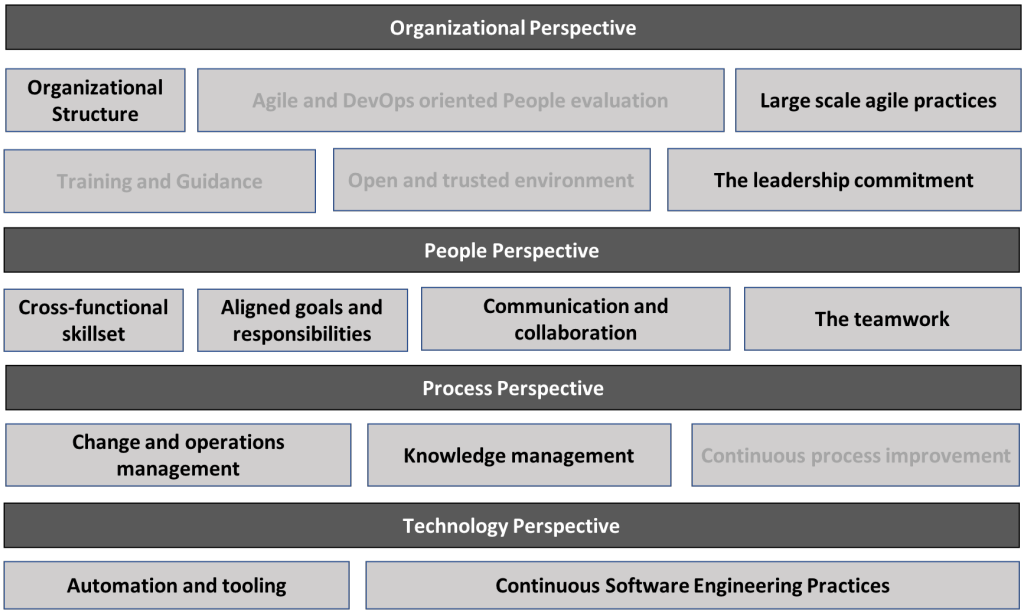

FIGURE 5.10: DevOps implementation framework variation highlighting the focus areas specific for Driver 2: Efficient value delivery to customers

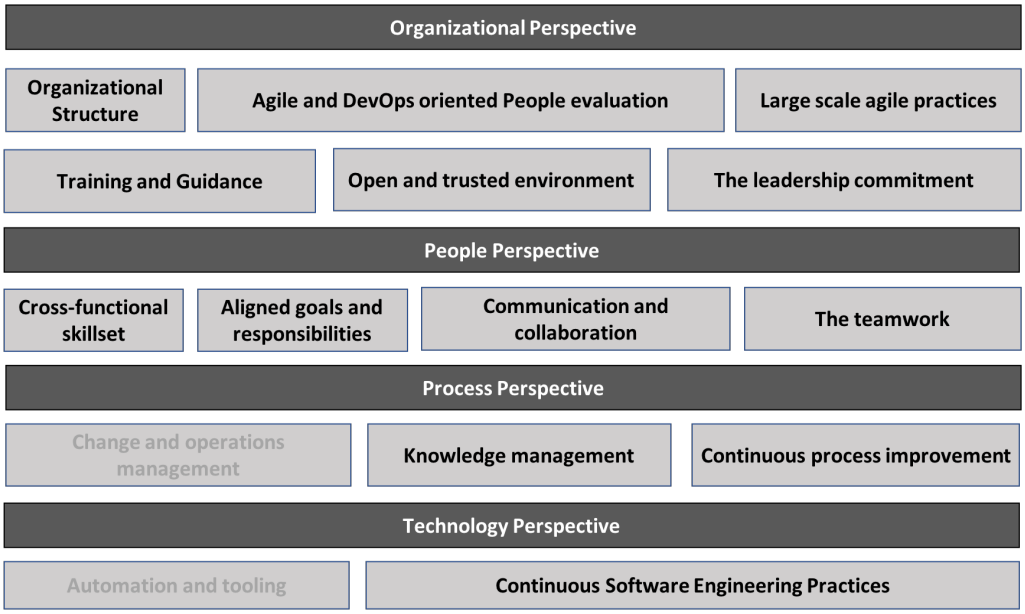

FIGURE 5.11: DevOps implementation framework variation highlighting the focus areas specific for Driver 3: Cooperative culture

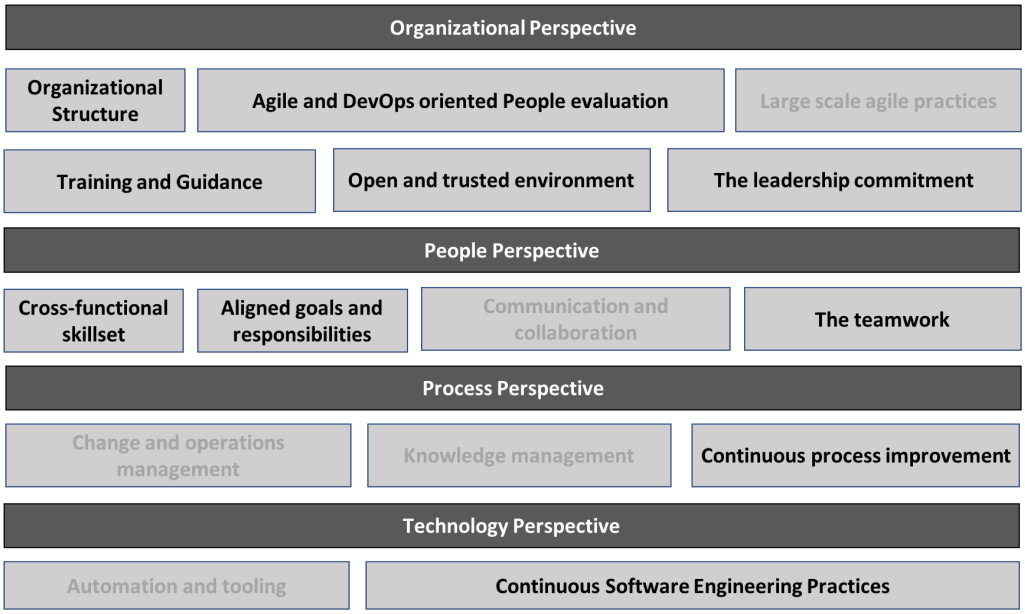

FIGURE 5.12: DevOps implementation framework variation highlighting the focus areas specific for Driver 4: Empowered people

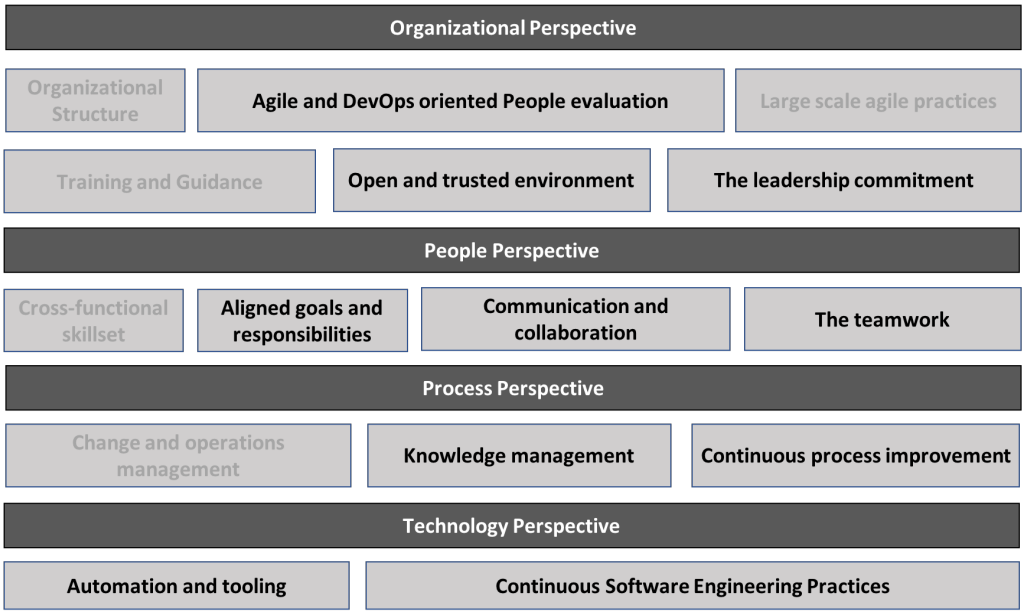

FIGURE 5.13: DevOps implementation framework variation highlighting the focus areas specific for Driver 5: Focus on continuous improvement

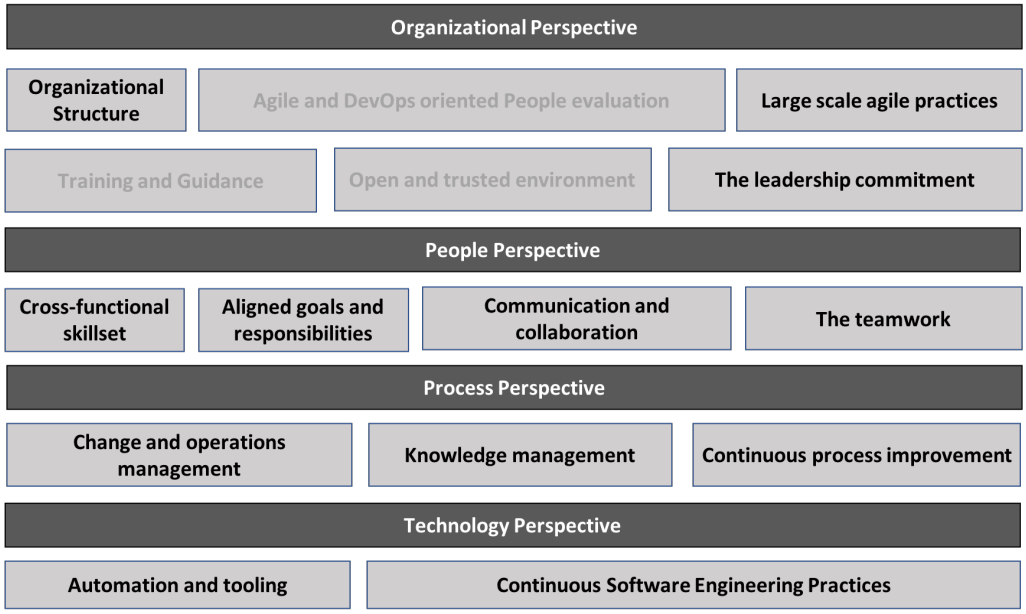

FIGURE 5.14: DevOps implementation framework variation highlighting the focus areas specific for Driver 6: Process and stakeholder alignment

6 Application of the artifacts to a real-world scenario

6.1 Role of measurement units and heuristics

Measuring the progress is an important part of an organization's transformation as they allow them to assess the factors that are influenced with the change process so that the necessary action can be taken when it is not progressing as expected. Forsgren and Kersten (2017) says that '[organizations] can not improve what they do not measure' and so measuring the DevOps transformation as well as the involved processes is quite important (Forsgren and Kersten 2017, p.2). Above all, one of the key principles of both Agile and DevOps is 'continuous improvement', which again stresses on perform-assess-improve cycle.

While it is significant to measure the transformation, it is more important to choose the right measurement units to perform the measurements. Because we get the details of what we measure. Thus it is important to measure all relevant perspectives among organizational, people, process and technology. Even several other authors agree with this approach of measuring the mentioned different perspectives (Ravichandran et al. 2016a; Swartout 2014). Therefore this section discusses about using the developed DevOps implementation framework and the identified drivers to delineate the selection of right measurement units, which would further be useful to select the right metrics later.

Before getting into the discussion of using the developed artifacts in the role of metrics selection, it is good to know the list of heuristics about metrics defined by Hartmann and Dymond since they are not only suitable for Agile but also for DevOps, considering DevOps as an extension of Agile (Hartmann and Dymond 2006). According to him, a good agile metric (in our case a good DevOps metrics as well) has the following characteristics:

- **Affirms and reinforces Lean and Agile (and DevOps) principles:** The selected metrics are not counterproductive but encourage the right behavior such as teamwork
- **Measures outcome, not output:** The delivered customer value is measured as the outcome.
- **Follows trends, not numbers:** The management considers the aggregated information and so not granular than the team level details.

- **Answers a particular question for a real person:** Metrics measure the worthwhile information
- **Belongs to small set of metrics and diagnostics:** Less number of metrics but enough information will be collected to get the required big picture.
- **Is possible to collect:** The relevant data is obtainable.
- **Reveals, rather than conceals, its context and significant variables:** Metrics are well explained so that all the influencing factors are visible to avoid the false assumptions.
- **Provides fuel for meaningful conversation:** The considered metrics encourage people to discuss about it.
- **Provides feedback on a frequent and regular basis:** The relevant metrics should be applied to take the measurements periodically and so continuous improvement will be possible.
- **May measure value or process:** Metrics measure anything that is suspected to be a factor affecting the effectiveness.
- **Encourages 'good enough' quality:** The metrics allow the business counterparts to define the expected quality so that the customer satisfaction can be made sure.

6.1.1 Using the DevOps implementation framework to identify the right measurement units

Every organization has a goal that reflects what are they trying to achieve by implementing DevOps within their organization. The given figure.6.1 shows how the organizational goal can be connected to developed artifacts, drivers, perspectives and focus areas from DevOps implementation framework and the relationship identified between them. As depicted in that figure, the identified focus areas and perspectives corresponding to the driver can be of help to identify the relevant measurement units which help to measure the progress towards the achievement of goals.

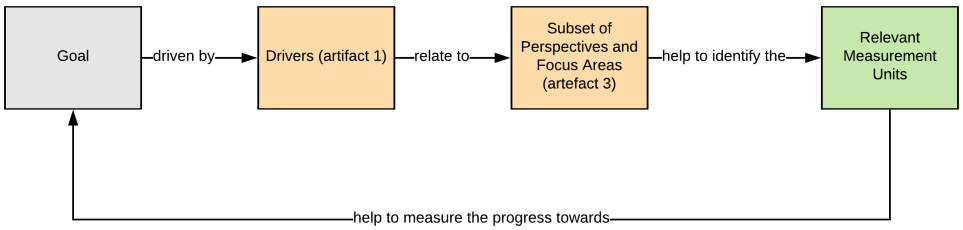

FIGURE 6.1: Generic diagram representing the relationship among goals, drivers, focus areas and measurement units

Sample 1: Goal - Faster delivery to the customer

One of the organizational goals of a Dutch financial organization, which participated in the study has a goal of expediting their process so that their customers can

enjoy their service and products faster than earlier. In order to explain the derivation of suitable measurement units, the goal has been taken in two ways: (a) from the people and organizational perspective and (b) from the process and technology perspective as shown in figure.6.2.

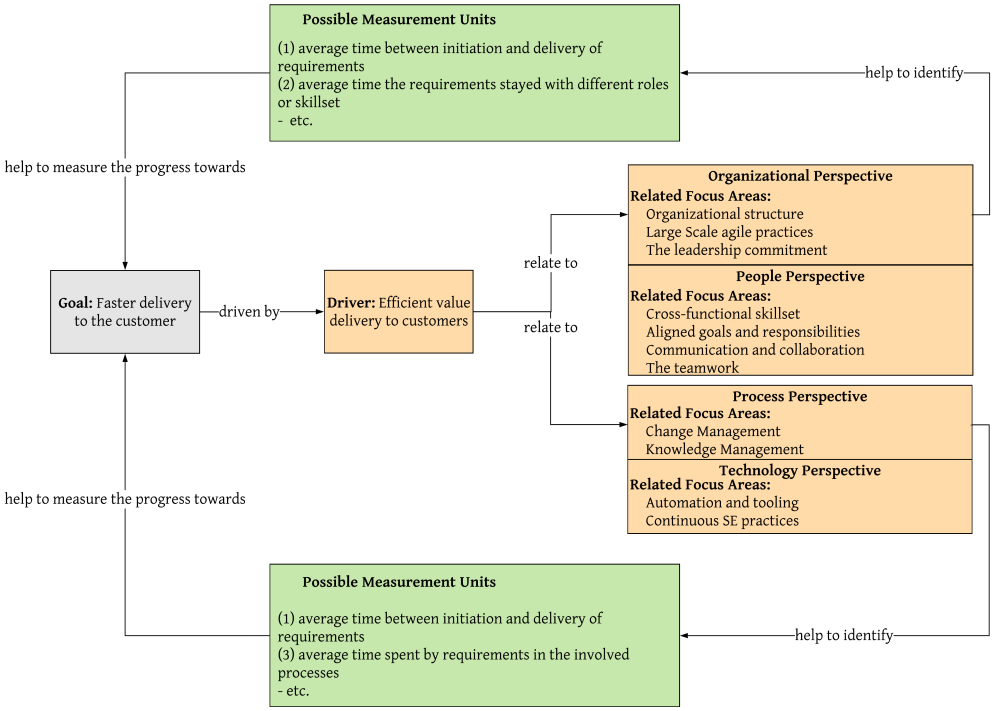

FIGURE 6.2: Relationship among goals, drivers, focus areas and measurement units specific to sample goal 1

In order to improve the time taken for delivery, it is important to first know how much time it currently takes for any requirement including new feature related requirements and change requests. Thus it is relevant to measure the (1) **time that takes between the initiation and the actual delivery of those requirements**. However, it might not really be enough to keep looking at the overall time that is being taken for the life-cycle of a requirement when the time stays indifferent. In that case, we can have a deeper look into the time by checking it in two different ways, which makes the given figure.6.2 to get separate branches into people and process perspectives.

From the people perspective, the requirements can be checked to see (2) **the time period that a requirement stayed with different roles**. It is useful to link the organizational perspective with people perspective so that the following questions can arise:

1. why does that specific role keep these requirements longer?
2. What is the average time spent by that role on other requirements?
3. what can be done to reduce the time spent by that role?

4. is this a common scenario with anyone taking that role or is it something specific to the person who took that role?

On the other hand, from the process and technology perspective, the requirements can be checked to see (3) **the time periods that a requirement stayed with involved processes such as development or functional testing** etc. This helps to identify which process takes longer which in turn initiates an analysis such as,

1. why that specific process takes longer than others for a requirement?
2. what is the average time spent on that process for other requirements?
3. how can that process be improved to reduce time?
4. is the improvement required on the identified process or any other dependent process? and
5. is it really the process that needs improvement or the people who involved in it?

Asking such questions helps to identify where the obstacle is and how can that obstacle be removed. The more important note is that these analysis should always lead to the identification of metrics that provoke the discussion of improvement points in terms of people, process or technology but not blaming each other.

Knowing the relevant focus areas which are related to the corresponding driver helps to ask the relevant questions. Moreover, they can be of help to go on and check the next relevant focus area from different perspectives so that the obstacles indirectly related to the goal can be identified and so the measurement units can be adjusted to measure the right focal point that needs attention.

Sample 2: Goal - Zero downtime

An other organization goal of the same organization is maintaining zero downtime of their systems, which is a customer-centric behavior that focuses of providing the maximum availability of their services to their customers. Although the goal is not limited to process and technology perspectives, they are considered as the principal ones since the goal is related to the systems. Thus, the figure.6.3 depicts the considered perspectives related to the goal of zero downtime and the relevant driver, agility and customer-centricity.

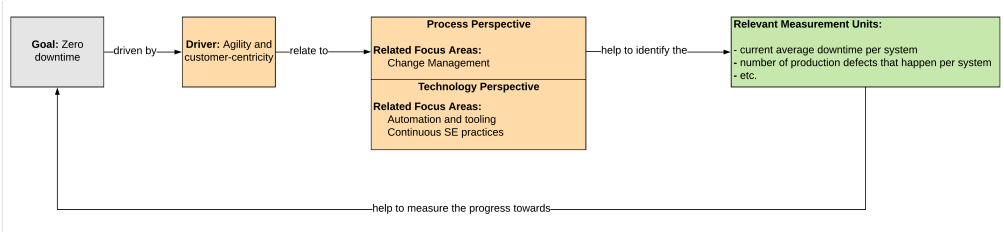

FIGURE 6.3: Relationship among goals, drivers, focus areas and measurement units specific to sample goal 2

As like with the previous sample goal, in order to bring the downtime of the systems to zero, the foremost important factor is to measure the (1)**current downtime**

i.e. the amount of time that the system is not available for its end-users. However, measuring only the downtime of the system is not useful by itself so it is also important to look into the causes for system unavailability. For example, the system may become unavailable due to the defects that occur in the system. If this is the case most often, then the organization should measure the (2) **number of production defects that happen with their systems** and analyze further to identify what can be improved in terms of people, process or technology to eliminate those defects in the future.

6.2 Sectional summary

This section first highlights the required characteristics of the measurement units taken from another literature. Then it applies the developed artifacts from previous section 5 into a couple of DevOps implementation goals $(\text{belong to one of the participating organizations})$ to demonstrate the role of artifacts in deriving the relevant measurement units.

7 Artifacts Validation

The expert opinion sessions were conducted with 5 experts who have various levels of experience working at financial organizations to evaluate the following developed artifacts:

1. DevOps implementation drivers of agile based financial organizations
2. DevOps implementation framework for agile based financial organizations
3. Relationship between the identified drivers and developed DevOps implementation framework

This section describes the evaluation scenario by providing background of the participated evaluators, the considered assessment criteria and also presents the evaluation results.

7.1 Artifacts Evaluation Scenario

7.1.1 Preparation and execution

Regarding the evaluation session, the following items have been prepared.

1. Powerpoint presentation for the session
2. Interview protocol (Appendix C)
3. Excerpt of the thesis report (Section 5), which provides the documentation of the developed artifacts

Due to the descriptive nature of the artifacts under evaluation, the evaluators' have been sent with the documentation of the artifacts and the interview protocol earlier to the evaluation session. This has allowed the evaluators to go through the documentation and helped them to be aware of the artifacts even before the session. According to my learning from the first session, the remain two session participants have been sent with the presentation slides as well in advance.

During the validation session, the evaluators have been presented with the artifacts one after the other and they have been asked with the criteria based questions related to the corresponding artifact. Moreover, they have been allowed to go through the printed documentation in order to get more details of any of the artifact whenever required. Having two evaluators in every session allowed them to discuss details with each other, which has enriched the evaluation results.

TABLE 7.1: Artifact evaluators' profile

Evaluation session #	Evaluator's profile	Evaluator's background
1	Delivery Manager	A veteran software professional with more than 25 years of experience in multiple positions and who currently leads multiple DevOps teams at a large Dutch bank in the Netherlands.
	DevOps team member	A senior team member and also an enthusiast of Agile and DevOps paradigms who facilitates ongoing DevOps transformations at a Dutch bank.
2	DevOps Architect	A DevOps architect who has 18+ years of working experience in the IT industry and in the financial domain. This person has consulting experience with DevOps implementation at several organizations.
	DevOps consultant	A business consultant with 13+ years of experience in the IT industry. This person has 7+ years of experience in the financial sector.
3	DevOps team member	A young professional specialized in Operations and currently being part of a DevOps team at a Dutch bank.

7.1.2 Profile of the evaluators

Since the developed artifacts are intended to be used for large scale agile based financial organizations, the participating experts are all invited from multiple large scale agile organizations within the Netherlands. All the participants are currently working for or had past experience at a financial organization and they have all contributed to or involved in at least one DevOps transformations so far. Moreover, the developed artifacts are designated to be used by all levels of employees including team members and managers. Thus, the expert profiles have been chosen to include both as it can be seen in table. 7.1.

7.1.3 Evaluation criteria

Evaluation of artifacts is an important phase besides artifact development in design science research (DSR) paradigm as they help to improve the artifacts iteratively before they get implemented in the context (Wieringa 2010). The interview protocol used for the validation of the current study artifacts can be found at Appendix C. The evaluation criteria have been identified from the hierarchy of IS artifact evaluation criteria developed by Prat et al. (2014). The following table 7.2 provides the list of evaluation criteria considered per artifact.

All the validation sessions were run with the same validation interview protocol (Appendix C) and so the list of criteria remained the same throughout the validation phase. The description of each of the considered criterion is given below.

- **Completeness:** This criterion assesses whether the artifact under assessment is complete and is not missing any components in its list (drivers, perspectives and focus areas).
- **Fit with organization:** Since the developed artifacts are not specific to any organization, this criterion is included to assess the suitability of the artifact for the participant's organization.
- **Understandability:** This criterion is included to assess whether the considered terms are coherent and their meaning is straightforward.

TABLE 7.2: Evaluation criteria per artifact

Artifact #	Artifact Description	Evaluation Criteria
1	DevOps implementation drivers of agile based financial organizations	Completeness Fit with organization Understandability
2	DevOps implementation framework for agile based financial organizations	Completeness Fit with organization Understandability Usefulness
3	Relationship between the identified drivers and developed DevOps implementation framework	Accuracy Usefulness

- **Usefulness:** It is the criterion that assesses how useful the artifacts are as found by the evaluators. It was followed by exploratory questions that inquired how and by whom the developed artifacts can be used within their organization.
- **Accuracy:** This criterion is used only for the last artifact and it is included to understand the evaluator's opinion on correctness of the relationship derived between the identified drivers and the developed DevOps implementation framework.

7.2 Evaluation Results

7.2.1 Artifact 1: DevOps implementation drivers of agile based financial organizations

The summary of evaluation results regarding the first artifact, drivers is given in the table 7.3.

TABLE 7.3: Evaluation results of artifact 1

Criterion	Session 1		Session 2		Session 3
	Evaluator 1	Evaluator 2	Evaluator 3	Evaluator 4	Evaluator 5
Completeness	++	++	++	++	++
Fit with organization	+-	+-	++	++	++
Understandability	++	++	++	++	++

++ Fully Agreed +- Partially Agreed - - Rejected

Completeness. The evaluators agreed that the identified list of drivers include most of the reasons why their organization want to implement DevOps. The evaluators from session 1 asked to look into the factor of 'cost' since their organization is looking at DevOps as a way to deliver value to their customers faster and also cheaper. According to this comment, an analysis has been made and the driver

'Faster value delivery to customer' has been changed into 'Efficient value delivery to customers' as shown in the following subsection 7.3. The other evaluators have been shown with the updated list of drivers and they all agreed with it.

Fit with organization. One of the drivers namely 'cooperative culture', was identified as not a suitable one for their organization by evaluator 1 although he agreed with the rest of the drivers (**iv-av-1**). However, he acknowledged that it can be a possible driver for other organizations. The other evaluators did agree that the the list of drivers are suitable to their organization.

Understandability. All the evaluators agreed that the terms given to the drivers were easy to understand and coherent to the context.

7.2.2 Artifact 2: DevOps implementation framework for agile based large financial organizations

The summary of evaluation results regarding the second artifact, DevOps implementation framework including the perspectives and focus areas is given in the table 7.4.

TABLE 7.4: Evaluation results of artifact 2

Criterion	Session 1		Session 2		Session 3
	Evaluator 1	Evaluator 2	Evaluator 3	Evaluator 4	Evaluator 5
Completeness	++	++	++	+-	++
Fit with organization	++	++	++	+-	++
Understandability	++	++	++	++	++
Usefulness	++	++	++	+-	++

++ Fully Agreed +- Partially Agreed - - Rejected

Completeness. Most of the evaluators agreed with the selection of the perspectives and the focus areas as seen with completeness score in table. 7.4. When the process perspective was concerned, some of the evaluators have mentioned that the 'change management' term is not complete for the corresponding focus area. This point was discussed during the first sessions as well but they did not ask for the change in the term since the given documentation is explaining about both the development and operations part. Thus, we have updated the focus area 'Change management' into 'Change and operations management' as shown in subsection. 7.3.

Fit with organization. Most of the evaluators agreed that the developed framework is suitable to their organization.

Understandability. The evaluators agreed that the terms used for all the perspectives and the focus areas are comprehensible.

Usefulness. When asked about the usefulness about the developed framework, most of them agreed that the given framework is somewhat useful for their organization. They have told that the framework is definitely useful for those who have not yet started with DevOps implementations but willing to do so. On the other hand, since the framework is much more on the high level, they have told

that the people working for mature Agile and DevOps organizations may not find it useful.

7.2.3 Artifact 3: Relationship between the identified drivers and developed DevOps implementation framework

The summary of evaluation results regarding the third artifact, the derived relationship between the drivers and the DevOps implementation framework by bringing out the framework variations is given in the table 7.5.

TABLE 7.5: Evaluation results of artifact 3

Criterion	Session 1		Session 2		Session 3
	Evaluator 1	**Evaluator 2**	**Evaluator 3**	**Evaluator 4**	**Evaluator 5**
Accuracy	not assessed	not assessed	++	++	++
Usefulness	+-	+-	++	+-	++

++ Fully Agreed +- Partially Agreed - - Rejected

Accuracy. The derived mapping of drivers and the DevOps implementation framework components' have been checked by the evaluators from sessions 2 and 3. They have agreed with the derived mapping based on the given rationalization. During the first session, this has not been assessed due to the shortage of time.
Some of the evaluators have told that although they agree with most of the given justifications, they said that the supporting facts are based on a small dataset. Therefore, they said that the accuracy of the derived relationships can not be fully agreed since they may be different when the large dataset is involved.

Usefulness. When asked about the usefulness of the final deliverable, few of the evaluators have got puzzled. Later knowing the possibility of using this deliverable to derive the measurement units, they have agreed that this can be useful for them. Some evaluators have told us that this can be useful for the metrics identification before we could tell them about the same.
However, some evaluators have told that the generic nature of the developed framework leads to a lot of overlap among the artifact 3 derivations. This overlapping limits the actual usefulness of this deliverable.

7.3 Evolution of the artifacts

7.3.1 Artifact 1: DevOps implementation drivers of agile based financial organizations

The following figures 7.1 and 7.2 depict the initial and updated versions of the DevOps implementation drivers identified for agile based financial organizations respectively. The updated version 7.2 has been developed based on the comments received from the evaluators of the first validation session. They mentioned that the given drivers miss the factor of cost as their organization see DevOps as a way to perform their work in a cost-effective way. To authenticate the applicability of the

received comment, the scientific literature has been approached. According to the other studies as well, DevOps is indeed considered as a way to do develop and deliver software in a faster and cheaper way (Wouter J.W. Geurts 2001). Therefore, we have renamed the second driver as shown in figure. 7.2 and adjusted the rest of the documentation and other artifacts accordingly.

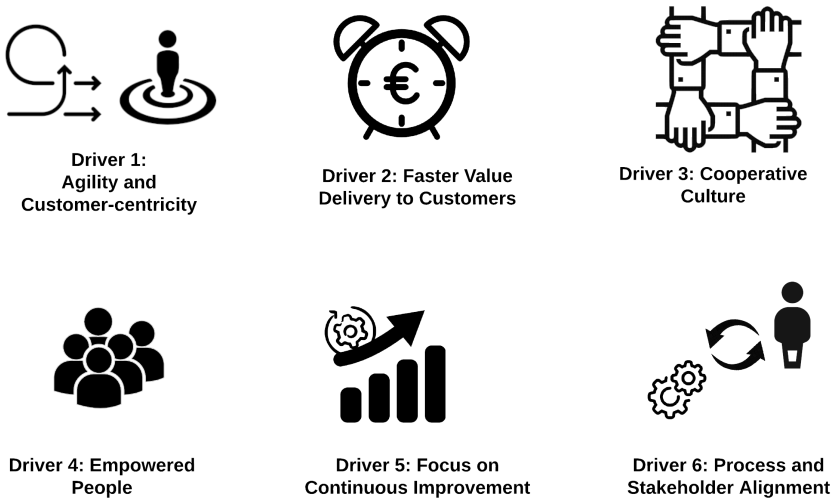

FIGURE 7.1: Initial version of identified drivers

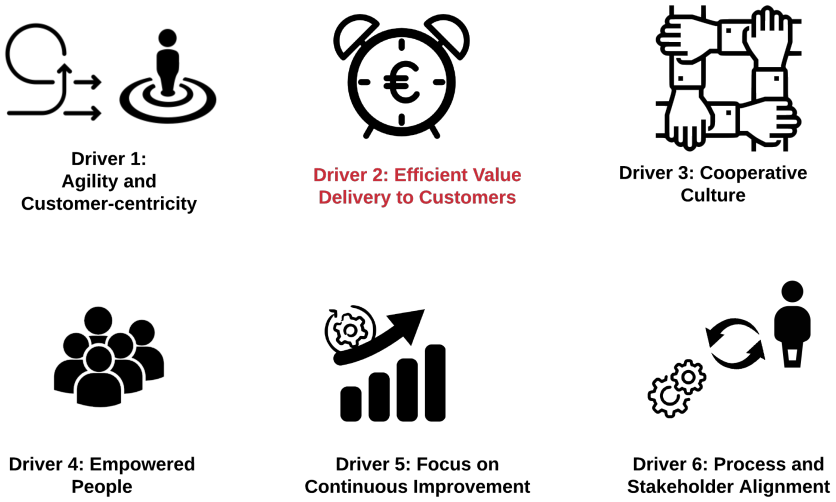

FIGURE 7.2: Updated version of identified drivers after validation session 1

Artifact 2: DevOps implementation framework for agile based large financial organizations

The given figures 7.3 and 7.4 show the corresponding versions of the second artifact before and after update. The latter has been developed after validation session 2 and

checked during session 3. According to them, the earlier given term 'Change management' sounds specific to the developmental activities regarding the introduction of a change. However, in an actual DevOps environment, the operational activities are also considered much more important since they make sure that the new changes do not negatively impact the ongoing business activities. Thus, to comply with that we have renamed the corresponding focus area as 'Change and operations management' and also updated the other parts of the document wherever needed.

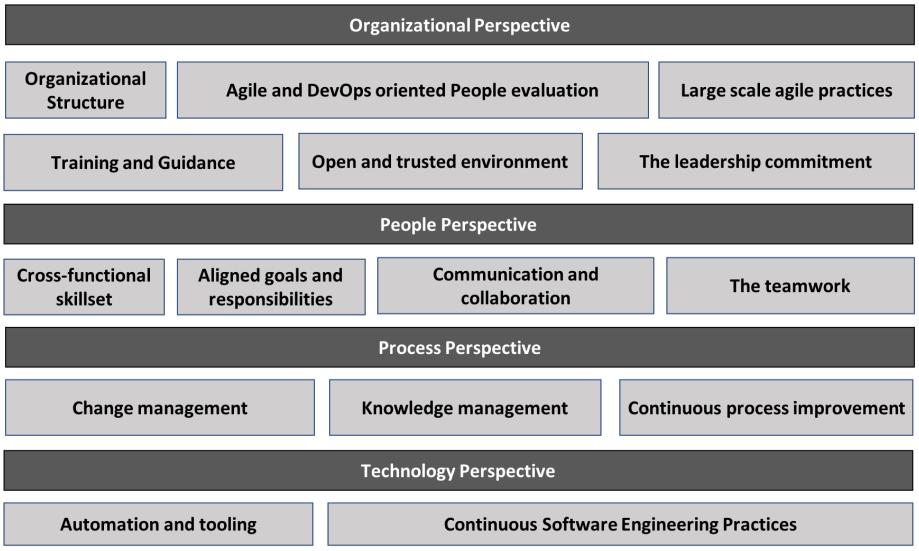

FIGURE 7.3: Initial version of the artifact 2 - DevOps implementation framework

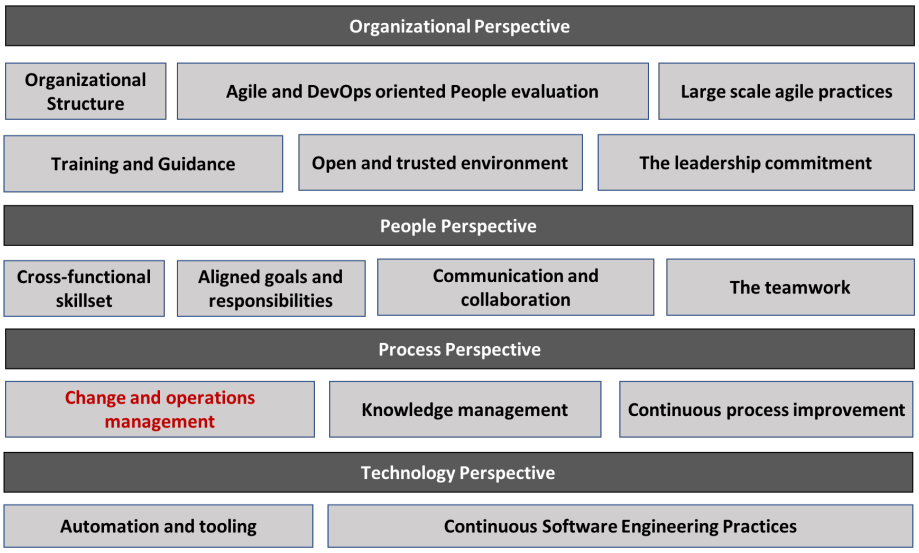

FIGURE 7.4: Updated version of the artifact 2 - DevOps implementation framework

7.4 Summary of the artifact validation

This validation section narrates the evaluation scenario in section 7.1 by indicating how the validation session have been run, the profile of the participants, and the

considered evaluation criteria. Next, in section 7.2, the evaluation results are reported per artifact. Finally, section 7.3 documents the changes done in the artifacts according to the comments received during the evaluation sessions.

8 Results and Discussion

8.1 Results

In order to answer the main research question, "How to design a DevOps implementation framework that is helpful for Agile based financial organizations?", we have developed different artifacts by answering several sub-research questions by following the design science framework. Thus, the sub-research questions are answered first and then the main research question is answered later.

SQ1: Why are financial organizations interested in implementing DevOps besides following Agile principles within their software development units?
Every organizational transformation should be well-thought and thus it should have strong reasons why the transformation is needed after all. These reasons provide the foundations for what an organization is trying to achieve in terms of goals. Therefore, it is important to understand the general reasons for Agile and DevOps transformations with any organization and these are summarized in Chapter 3 (subsections 3.1.1 and 3.2.2), results of the literature review. However, the specific reasons for financial organizations have been understood from the practitioner interviews and this has resulted as the first artifact - Artifact 1: Main drivers behind Agile and DevOps adoption at financial organizations (subsection 5.1). This artifact identifies the following six primary drivers.

- Driver 1: Agility and customer-centricity
- Driver 2: Efficient value delivery to customers
- Driver 3: Cooperative culture
- Driver 4: Empowered people
- Driver 5: Focus on continuous improvement
- Driver 6: Process and stakeholder alignment

SQ2: What are the different perspectives and the main focus areas that are required for a DevOps implementation while following Agile principles at financial organizations?
From the literature study that refers to subsections 3.1.2 and 3.2.3 (Chapter 3), it is understood that there are several factors that are involved in the Agile and DevOps transformations of an organization and these factors play a crucial role in the success of such transformations. Based on what these factors are about or who is responsible to those factors, the agile implementation factors are categorized into different perspectives by different authors. However, it is understood from the practitioner

interviews that these perspectives still hold true for DevOps transformational factors as well. Therefore, the following perspectives are deduced from literature and utilized in this study. The details about these perspectives and the relationship considered among these perspectives can be found at subsection 5.2.1 (Chapter 5).

- Organizational perspective
- People perspective
- Process perspective
- Technology perspective

Knowing the perspectives from the literature, the main factors within every perspective that determines the success of the DevOps transformation has been identified from the practitioner interviews. These factors are mentioned as 'focus areas' since these are the factors that specifically need attention towards the success of the transformation but they are not the only factors involved in the DevOps way of working. There are 15 such focus areas identified and every focus area is associated to one of the given four perspectives. The more details about the focus areas like, what a specific focus area is about, why is it considered as important for the DevOps implementation and which perspective does the focus area belong to and why are answered within the documentation of every focus area at subsection 5.2 (Chapter 5).

SQ3: How does a conceptual framework incorporating identified perspectives and focus areas look like?
After identifying the perspectives and the corresponding focus areas, they have been put together into one conceptual framework, which is shown in subsection 5.3. In his journal article, Jabareen defines a conceptual framework as a 'plane of interlinked concepts that together provide a comprehensive understanding of a phenomenon' (Jabareen 2009, p.51). In our case, DevOps is the studied phenomenon, and the developed framework highlights the relevant concepts in the name of perspectives and focus areas.

SQ4: How can the relationship between identified drivers and the developed DevOps implementation framework be revealed?
In order to identify the relationship between the identified drivers (artifact 1) and the developed framework, in terms of perspectives and focus areas, every pair of driver - focus area combination has been investigated. Both the practitioner interviews and the literature studies have been probed for this activity. As a result, the relationship between every possible combination of the driver and focus area is justified within the corresponding focus area documentation given in subsection 5.2 (Chapter 5).

SQ5: How to identify and relate the measurement units to the drivers and the developed framework?
This question is aiming to give a direction to the organizations how they can use the developed artifacts of this study to identify the measurement units (a.k.a. metrics) that can measure their progress towards their DevOps transformational goals. In order to explain that, we have considered the real organizational goals of one of

the participating financial organizations and explained how the relationship among the goals, identified drivers, perspectives and focus areas of developed framework may influence the identification of helpful measurement units that can measure the progress towards the corresponding goals in Chapter 6.

Main Research Question: "How to design a DevOps implementation framework that is helpful for Agile based financial organizations?"
From the question, it is obvious that the DevOps implementation framework is supposed to be developed and it has been developed by answering the sub-research questions 2 and 3 (SQ2 and SQ3). However, the intention was not to create yet another framework but to create one that is **useful** for the financial organizations, which has interest in agile software development methods. Thus, we have identified the drivers (SQ1), developed the variations of the framework corresponding to the identified drivers (SQ4) and also explained a real-time scenario, where the developed artifacts can be applied (SQ5). The developed conceptual framework is generic in nature so that it can serve as a basic guideline for anyone who participates in a DevOps implementation and the evaluation results are acknowledging the same. Moreover, we do want to highlight this process of involving the drivers and measurement units to the development of the DevOps implementation framework and thus the given main research question is formed as such. We believe that this question has been rightly answered by combining answers to all the sub research questions.

8.2 Discussion

The current study has developed a high-level framework that encompasses the various perspectives and the focus areas that are relevant for the successful DevOps implementation of a large-scale organization. To maximize the usefulness of the framework and to make it specific for the financial organizations, from where the practitioners are selected to participate in this study, we have also identified the drivers and we have demonstrated the derivation of measurement units based on the DevOps implementation goals.

This study is comparable to other studies which identified the success factors of an Agile implementation from various perspectives namely Chow and Cao (2008), Darwish and Rizk (2015), and Hameed et al. (2016). However, while the mentioned the studies had concentrated on Agile, the current one has concentrated on a DevOps implementation where Agile is also followed. This study expects the involved organization to already follow Agile or to implement Agile along with the DevOps implementation.

Next, we want to compare this study with other DevOps maturity models that we have reviewed in the section 3 of this report. While comparing this study with the other mentioned studies viz. Mohamed (2016), Bucena and Kirikova (2017), and Feijter et al. (2018), the following differences can be observed: none of the mentioned studies are specific to the financial industry; the current study's framework

can be expanded to an 'organization specific maturity model' in which every focus area is defined with the list of capabilities that the involving organization want to progressively reach; on the contrary, the mentioned studies have developed a generic maturity model, which may not be suitable to every organization but they may not be suitable to be tailored; none of the mentioned studies have revealed their model's connection with the DevOps implementation goals and measurement units as the current study has done.

8.3 Limitations of the study

External Validity. This refers to the extent to which the current study results can be generalized to other population and settings. All the participating organizations within the problem investigation and data collection phase are a single type of financial organizations and they are comparably similar in size. On the other hand, the evaluation phase includes people from both financial and non-financial organizations but the participants have experience with Agile and DevOps adoption at financial organizations. However, all the participants are personally interviewed during the study and they are all based in the country of Netherlands. Although the artifacts are developed not only based on the interview data but also according to the data collected from scientific literature studies, specifically the artifact 1 (drivers) and the artifact 3(DevOps implementation framework variations per driver) have higher chances of affected by this validity errors. Thus it is possible that the other studies that involve different type of financial organizations from another geographical area may result dissimilar outcome.

Internal validity. The internal validity is the measure that is related to the cause and effect related errors. By the time of this study, the participating organizations are undergoing implementations of Agile and DevOps and so their Agile and DevOps implementations are in various maturity levels. Therefore, it is difficult to identify the factors that are influenced only due to DevOps implementation but not by Agile implementation or vice versa. Hence, there is a possibility that another study may result in unlike artifacts due to their Agile and DevOps maturity level and implementation history.

Face validity. Face validity attempts to measure how far a research study meets its objectives. As mentioned in the earlier sections of the document, this study attempts to bring up a DevOps implementation framework that is useful for people within those agile based financial organizations to bring them awareness, providing a base with their DevOps implementations and shows how to identify the appropriate measurement units useful for their goal oriented DevOps implementation. In order to measure how far the developed artifacts are perceived as useful and understandable, the criteria based evaluation sessions have been conducted with industrial experts as explained in chapter 7. However, the face validation is performed only with a few experts who are all from the Netherlands. This may not be enough to prove the usefulness of the artifacts, which may be necessary for the actual adoption of the framework for the practical purpose.

9 Conclusions

9.1 Research Contributions

This master thesis has developed certain artifacts, which are suitable for large-scale agile based financial organizations in the context of DevOps implementation. The implications of this study to the scientific community and to the practitioners in this field are summarized as follows.

9.1.1 Scientific contribution

The developed DevOps implementation framework provides a high-level overview of the involved perspectives and relevant focus areas within those perspectives. As reviewed in section 3, there are similar models available for Agile implementation but not for DevOps adoption. Moreover, there is no specific DevOps adoption framework available for financial organizations and thus the following scientific contributions are observed from this study.

- Development of artifacts such as drivers, DevOps implementation framework and its variations per driver specific for agile-based large-scale financial organizations
- Revealing the process of relating a framework with their drivers and measurement units

While the first contribution highlights the developed artifacts specific to a user group, the latter contribution guides any other researcher focusing on developing an implementation framework to relate it to the drivers (specifies why such an implementation is required) and the measurement units (specifies how they can contribute towards measuring the implementation progress).

9.1.2 Societal contribution

From the evaluation session results that is described in section 7, most of the practitioners find the artifacts to be useful for their organizational scenario. The framework has been kept with high-level components so that it can be convenient to educate the people within the target organizations about the DevOps implementation, and it is suitable to the organizations that already follow Agile or that want to implement DevOps along with Agile. Furthermore, section 6 of this document highlights a scenario in which, the measurement units specific to the DevOps implementation goals are being derived with the help of developed artifacts.

Although the study has certain limitations as discussed in section 8.3, we believe that these contributions are still worthwhile to a certain extent.

9.2 Future Research

From the evaluation results and the discussed limitations, there are several possibilities identified to improve and extend this research as discussed here.

First of all, most of the people participated in the data collection as well as evaluation of this study are having experience from large banks, one among the several types of financial organizations. Thus, the future studies need to concentrate on performing the research by involving people from non-bank financial institutions such as insurance companies. Comparing the current study with another study focusing on non-bank financial institutions may bring interesting results.

Secondly, as per one of the evaluators, the current study presents focus areas in a very generic way such that the differences among the framework variations per driver are very less. This seems to limit the usefulness of the developed framework variations in a real-time implementation. Thus, the future studies are recommended to develop framework variations such that the focus areas are more specific towards achieving the corresponding driver with less overlap among other variations.

Thirdly, the current study artifacts have been evaluated with the expert opinion approach, which limits the validation intensity. Furthermore, we tried to apply the artifacts to a real organization's DevOps implementation goals to justify their usage in a real world scenario. However, in order to measure the actual usefulness of the artifacts and to improve them further, the technical action research should be used as the validation method. The latter helps to test the developed artifacts by utilizing them in a real financial organization's DevOps implementation project so that the concrete issues with the developed artifacts can be identified and further improved.

Finally, we do recommend to customize the current high-level framework by filling its focus areas' content with the internal best practices of an enterprise and then utilize the same for new DevOps implementations within the enterprise. In this way, the developed framework can be used as a DevOps implementation knowledge management aid, which helps to gather, share, utilize and update the DevOps implementation best practices of a large-scale organization.

9.3 Report Summary

As like with any other industry, DevOps is becoming popular among financial software organizations. Because of the advantages observed with Agile software development methods such as faster development time, improved quality and high customer-satisfaction, several large-scale financial organizations prefer Agile methods over traditional software development methods like waterfall software development method. However, it is not still not clear why they are interested in implementing DevOps along with or on-top of agile implementation. Thus, this thesis concentrates on identifying the drivers for large-scale financial organizations to go

for DevOps along with agile software development method. Nevertheless, with DevOps adoption, several existing factors get affected and many other new factors need attention. Thus, the current study brings up a framework based on the high-level factors from different perspectives that are required for the DevOps implementation in such organizations and develops the variations of the framework based on its relationship with the identified drivers. Later, to justify the usage of the developed artifacts - drivers, DevOps implementation framework and its variations, an application scenario with a real organization's goals have been presented.

Bibliography

Abidin, F. A. Z., Jawawi, D. N. A., and Ghani, I. (2017). Agile Transition Model Based on Human Factors. *International Journal of Innovative Computing* 7(1), 23–32.

Adel, A. and Abdullah, B. (2015). A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues* 12(1), 106–111. ISSN: 1694-0784.

Almomani, M. A., Basri, S., and Kamil, A. (2018). Factors Influencing the Implementation of Software Process Improvement in Small and Medium Enterprises: An Empirical Study. *Advanced Science Letters* 24(3), 1716–1722. ISSN: 1936-6612.

Alqudah, M. and Razali, R. (2016). A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology* 6(6), 828–837. ISSN: 2460-6952.

Augustine, B. S., Payne, B., Sencindiver, F., and Roject, A. G. P. (2005). M A N A G E M E N T : STEERING. 48(12), 85–89.

Babar, Z., Lapouchnian, A., Yu, E., Babar, Z., Lapouchnian, A., Yu, E., Devops, M., Choices, D., and Process, U. (2017). Modeling DevOps Deployment Choices Using Process Architecture Design Dimensions To cite this version : HAL Id : hal-01442259 Modeling DevOps Deployment Choices using Process Architecture Design Dimensions.

Bang, S. K., Chung, S., Choh, Y., and Dupuis, M. (2013). A grounded theory analysis of modern web applications. *Proceedings of the 2nd annual conference on Research in information technology - RIIT '13* (October 2013), 61.

Bessant, J., Caffyn, S., and Gallagher, M. (2001). An evolutionary model of continuous improvement behaviour. *Technovation* 21(2), 67–77. ISSN: 01664972.

Bhadoriya, N., Mishra, N., and Malviya, A. (2014). Agile Software Development Methods, Comparison with Traditional Methods & Implementation in Software Firm. 3(7), 1656–1662.

Brown, A. W., Ambler, S., and Royce, W. (2013). Agility at scale: Economic governance, measured improvement, and disciplined delivery. *Proceedings - International Conference on Software Engineering*, 873–881. ISSN: 02705257.

Bucena, I. and Kirikova, M. (2017). Simplifying the DevOps Adoption Process. In: *International Conference on Perspectives in Business Informatics Research*. Copenhagen, Denmark, 1–15.

Chatman, J. A. and Barsade, S. G. (1995). Personality , Organizational Culture , and Cooperation : Evidence from a Business Simulation Author (s): Jennifer A . Chatman and Sigal G . Barsade Source : Administrative Science Quarterly , Vol . 40 , No . 3 (Sep ., 1995), pp . 423-443 Published by. *Administrative Science Quarterly* 40(3), 423–443.

Chen, L. (2015). Continuous delivery: Huge benefits, but challenges too. *IEEE Software* 32(2), 50–54. ISSN: 07407459.

Chow, T. and Cao, D.-b. (2008). A survey study of critical success factors in agile software projects. *The Journal of Systems & Software* 81, 961–971.

Cockburn, A. and Highsmith, J. (2001). Agile software development: The people factor. *Computer* 34(11), 131–133. ISSN: 00189162.

Conboy, K., Coyle, S., Wang, X., and Pikkarainen, M. (2011). People over process: Key challenges in agile development. *IEEE Software* 28(4), 48–57. ISSN: 07407459.

Darwish, N. R. and Rizk, N. M. (2015). Multi-Dimensional Success Factors of Agile Software Development Projects. *International Journal of Computer Applications* 118(15), 23–30.

Davis, J. and Daniels, K. (2015). *Effective DevOps*. Vol. 53. 9. Sebastapol, CA: O'Reilly Media, Inc., 1–155. ISBN: 9788578110796.

De Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., and Bijlsma, L. (2006). On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support. In: *2006 International Workshop on Software Product Management (IWSPM'06 - RE'06 Workshop)*. Minneapolis/St. Paul, MN, USA: IEEE, 3–12.

Demirkan, H. and Spohrer, J. (2015). T-Shaped Innovators. *Research Technology Management* 58(5), 12–15. ISSN: 08956308.

Diebold, P. and Dahlem, M. (2014). Agile Practices in Practice - A Mapping Study. In: *International Conference on Evaluation and Assessment in Software Engineering (EASE' 14)*. May. London, England: ACM, 1–11. ISBN: 9781450324762.

Dingsøyr, T. and Moe, N. B. (2014). Towards Principles of Large-Scale Agile Development - A Summary of the Workshop at XP2014 and a Revised Research Agenda. *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation* Volume 199, 1–8. ISSN: 18651348.

Dyck, A., Penners, R., and Lichter, H. (2015). Towards definitions for release engineering and DevOps. In: *3rd International Workshop on Release Engineering (RELENG 2015)*. Piscataway, NJ, USA: IEEE, 3. ISBN: 9781479919345.

Elberzhager, F., Arif, T., Naab, M., Sï¿½ï¿½, I., and Koban, S. (2017). From agile development to devops: Going towards faster releases at high quality - Experiences from an industrial context. In: *Lecture Notes in Business Information Processing*. ISBN: 9783319494203.

Elliot, S. (2015). DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified. *IDC Insight* (December).

Erich, F. M. A., Amrit, C., and Daneva, M. (2017). A Qualitative Study of DevOps Usage in Practice. *Journal of Software: Evolution and Process* 29(6), e1885. ISSN: 20477481.

Erich, F., Amrit, C., and Daneva, M. (2014a). A mapping study on cooperation between information system development and operations. In: *International Conference on Product-Focused Software Process Improvement (PROFES 2014)*. Ed. by A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch, and M. Raatikainen. Vol. 8892. 1. Cham, Switzerland: Springer, 277–280. ISBN: 978-3-319-13835-0.

Erich, F., Amrit, C., and Daneva, M. (2014b). *DevOps Literature Review*.

Farid, A. B. (2017). Enhancing Lean Software Development by using DevOps Practices. 8(7), 267–277.

Feijter, R. de, Vliet, R. van, Jagroep, E., Overbeek, S., and Brinkkemper, S. (2017). Towards the adoption of DevOps in software product organizations: A Maturity Model Approach. PhD thesis. Utrecht University, 1–173.

Feijter, R. D., B, S. O., Vliet, R. V., and Jagroep, E. (2018). DevOps Competences and Maturity for Software Producing Organizations. In: *Lecture Notes in Business Information Processing*. Vol. 318. Springer International Publishing, 244–259. ISBN: 9783319917047.

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

Fontana, R. M., Fontana, I. M., Da Rosa Garbuio, P. A., Reinehr, S., and Malucelli, A. (2014). Processes versus people: How should agile software development maturity be defined? *Journal of Systems and Software* 97, 140–155. ISSN: 01641212.

Forsgren, N. and Kersten, M. I. K. (2017). DevOps Metrics. *ACM Queue* (december), 1–16. ISSN: 0001-0782.

França, B. B. N. de, Jeronimo, H., and Travassos, G. H. (2016). Characterizing DevOps by Hearing Multiple Voices. In: *Proceedings of the 30th Brazilian Symposium on Software Engineering (SBES '16)*, 53–62. ISBN: 9781450342018.

Franklin, M. (2014). A Practical Framework for Successful Change Planning and Implementation. In: *Agile change management*, 1–8.

Gandomani, T. J., Zulzalil, H., Ghani, A. A. A., Sultan, A. B. M., and Sharif, K. Y. (2014). How Human Aspects Impress Agile Software Development Transition and Adoption. *International Journal of Software Engineering and Its Applications* 8(1), 129–148. ISSN: 17389984.

Gatchalian, M. M. (1997). People empowerment: The key to TQM success. *TQM Magazine* 9(6), 429–433. ISSN: 0954478X.

Geurts, W. J. (2016). *Faster is Better and Cheaper*.

Ghobadi, S. and D'Ambra, J. (2012). Coopetitive relationships in cross-functional software development teams: How to model and measure? *Journal of Systems and Software*. ISSN: 01641212.

Gill, R. (2002). Change management - or leadership? *Journal of Change Management* 3(4), 307–318. ISSN: 14697017.

Hameed, T. A. E., Latif, M. A. E. L., and Kholief, S. (2016). Identify and Classify Critical Success Factor of Agile Software Development Methodology Using Mind Map. *International Journal of Advanced Computer Science and Applications* 7(5), 83–92.

Hartmann, D. and Dymond, R. (2006). Appropriate agile measurement: Using metrics and diagnostics to deliver business value. In: *Proceedings - AGILE Conference, 2006*. IEEE, 126–131. ISBN: 0769525628.

Hobbs, B. and Petit, Y. (2017). *Agile Methods on Large Projects in Large Organizations*. Vol. 48. 3, 3–19. ISBN: 0080074648490.

Hon, W. K. and Millard, C. (2016). *Use by Banks of Cloud Computing : An Empirical Study*.

Horney, N., Pasmore, B., and O'Shea, T. (2010). Leadership Agility: A Business Imperative for a VUCA World. *People & Strategy* 33(4), 32–38. ISSN: 1946-4606.

Hsieh, H.-F. and Shannon, S. E. (2005). Three Approaches to Qualitative Content Analysis. *Qualitative health research* 15(9), 1277–1288.

Huttermann, M. (2012). *DevOps for Developers*. First. Berkeley, CA, USA: Apress, 1–184. ISBN: 978-1-4302-4570-4.

Iden, J., Tessem, B., and Päivärinta, T. (2011). Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts. *Information and Software Technology* 53(4), 394–406. ISSN: 09505849.

Iivari, J. and Iivari, N. (2011). The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology* 53(5), 509–520. ISSN: 09505849.

Jabareen, Y. (2009). Building a Conceptual Framework: Philosophy, Definitions, and Procedure. *International Journal of Qualitative Methods* 8(4), 49–62. ISSN: 1609-4069.

Jabbari, R., Ali, N. bin, Petersen, K., and Tanveer, B. (2016). What is DevOps? A Systematic Mapping Study on Definitions and Practices. *Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops*, 1–11. ISSN: 07421222.

Jalali, S. and Wohlin, C. (2012). Systematic Literature Studies: Database Searches vs. Backward Snowballing. *ESEM'12: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 29–38. ISSN: 1938-6451.

Javdani, T., Zulzalil, H., Azim, A., Ghani, A., and Sultan, A. B. (2001). On the Current Measurement Practices in Agile Software Development. 9(4), 1–7.

Kouzari, E., Gerogiannis, V. C., Stamelos, I., and Kakarontzas, G. (2015). Critical Success Factors and Barriers for Lightweight Software Process Improvement in Agile Development - A Literature Review. *Proceedings of the 10th International Conference on Software Engineering and Applications*, 151–159.

Kupiainen, E., Mäntylä, M. V., and Itkonen, J. (2015). Using metrics in Agile and Lean software development - A systematic literature review of industrial studies. *Information and Software Technology* 62(1), 143–163. ISSN: 09505849.

Leppanen, M. (2013). Information Systems Development. In: *Information Systems Development: Reflections, Challenges and New Directions*. Ed. by R. Pooley. Vol. 35. New York: Springer Science + Business Media. Chap. 27, 329–343.

Liang, T.-P. and Tanniru, M. (2007). Special Section: Customer-Centric Information Systems. *Journal of Management Information Systems* 23(3), 9–15. ISSN: 0742-1222.

Lindsjørn, Y., Sjøberg, D. I., Dingsøyr, T., Bergersen, G. R., and Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software* 122, 274–286. ISSN: 01641212.

Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2015). Dimensions of DevOps. In: *XP: International Conference on Agile Software Development*. Vol. 212. Cham, Switzerland: Springer International Publishing, 212–217. ISBN: 9783319186115.

Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2016a). An Exploratory Study of DevOps : Extending the Dimensions of DevOps with Practices. In: *International Conference on Software Engineering Advances (ICSEA)*. Rome, Italy.

Lwakatare, L. E., Kuvaja, P., and Oivo, M. (2016b). Product-Focused Software Process Improvement. 10027, 399–415.

McGinnis, T. C. and Huang, Z. (2007). Rethinking ERP success: A new perspective from knowledge management and continuous improvement. *Information and Management* 44(7), 626–634. ISSN: 03787206.

Moe, N. B., Dingsøyr, T., and Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology* 52(5), 480–491. ISSN: 09505849.

Mohamed, S. I. (2015). DevOps Shifting Software Engineering Strategy Value Based Perspective. *IOSR Journal of Computer Engineering Ver. IV* 17(2), 2278–661.

Mohamed, S. I. (2016). DevOps Maturity Calculator DOMC -Value oriented approach. *International Journal of Engineering Research & Science* 2(2), 2395–6992.

Neely, S. and Stolt, S. (2013). Continuous delivery? Easy! Just change everything (well, maybe it is not that easy). *Proceedings - AGILE 2013*, 121–128.

Nybom, K. B., Smeds, J., and Porres, I. (2016). Agile Processes, in Software Engineering, and Extreme Programming. 251, 131–143. ISSN: 18651348.

Oates, B. J. (2005). *Researching Information Systems and Computing*, 70–201.

Parizi, R. M., Gandomani, T. J., and Nafchi, M. Z. (2014). Hidden facilitators of agile transition: Agile coaches and agile champions. In: *2014 8th Malaysian Software Engineering Conference, MySEC 2014*. ISBN: 9781479954391.

Parker, D. W., Holesgrove, M., and Pathak, R. (2015). Improving productivity with self-organised teams and agile leadership. *International Journal of Productivity and Performance Management* 64(1), 112–128. ISSN: 1741-0401.

Phifer, B. (2011). Next-generation process integration: CMMI and ITIL do devops. *Cutter IT Journal* 24(8), 28–33. ISSN: 15227383.

Pinto, M. B., Pinto, J. K., and Prescott, J. E. (1993). Antecedents and Consequences of Project Team Cross-Functional Cooperation. *Management Science* 39(10), 1281–1297. ISSN: 0025-1909.

Prat, N., Comyn-Wattiau, I., and Akoka, J. (2014). Design-Science Research – a Holistic View. In: *18th Pacific Asia Conference on Information Systems, Chengdu, China.* 16.

Quinn, R. E. and Spreitzer, G. M. (1997). *The Road to Empowerment - 7 Questions Every Leader Should....pdf*.

Ravichandran, A., Taylor, K., and Waterhouse, P. (2016a). *DevOps for Digital Leaders*. Second edi. Berkeley, CA: Apress. ISBN: 978-1-4842-1841-9.

Ravichandran, A., Taylor, K., and Waterhouse, P. (2016b). DevOps for Digital Leaders.

Rigby, D. K., Sutherland, J., and Takeuchi, H. (2016). *Embracing agile*.

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., and Männistö, T. (2016). DevOps adoption benefits and challenges in practice: A case study. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 590–597. ISBN: 9783319490939.

Savor, T., Douglas, M., Gentili, M., Williams, L., Beck, K., Stumm, M., Facebook, T. S., Way, H., Douglas, M., Gentili, M., Williams, L., Facebook, K. B., and Stumm, M. (2016). Continuous Deployment at Facebook and OANDA. In: *38th International Conference on Software Engineering Companion (ICSE '16)*. New York, USA: ACM, 21–30. ISBN: 9781450342056.

Seo, D. and Paz, A. I. L. (2015). Exploring the Dark Side of IS in Achieving Organizational Agility. *Communications of the ACM* 51(February), 136–139.

Serrador, P. and Pinto, J. K. (2015). Does Agile work ? — A quantitative analysis of agile project success. *International Journal on Project Management* 33(5), 1040–1051. ISSN: 0263-7863.

Shahin, M., Zahedi, M., Babar, M. A., and Zhu, L. (2017). Adopting Continuous Delivery and Deployment. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering - EASE'17* (May), 384–393.

Sharma, S. (2017). *The DevOps Adoption Playbook: A Guide to adopting DevOps in a multi-speed IT Enterprise*. ISBN: 9781119308744.

Sherehiy, B., Karwowski, W., and Layer, J. K. (2007). A review of enterprise agility: Concepts, frameworks, and attributes. *International Journal of Industrial Ergonomics* 37(5), 445–460. ISSN: 01698141.

Smeds, J., Nybom, K., and Porres, I. (2015). DevOps: A Definition and Perceived Adoption Impediments. In: *International Conference on Agile Software Development (XP 2015)*. Ed. by C. Lassenius, T. Dingsøyr, and M. Passivaara. Vol. 212. Cham, Switzerland: Springer International Publishing, 166–177. ISBN: 9783319186115.

Stahl, D., Martensson, T., and Bosch, J. (2017). Continuous practices and devops: beyond the buzz, what does it all mean? *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (December), 440–448.

Stillwell, M. and Coutinho, J. G. F. (2015). A DevOps approach to integration of software components in an EU research project. *Proceedings of the 1st International Workshop on Quality-Aware DevOps - QUDOS 2015* (c), 1–6.

Swartout, P. (2014). *Continuous Delivery and DevOps: A Quickstart guide, Second Edition*. Second Edi. Birmingham, UK: Packt Publishing, 1–196. ISBN: 9781849693684.

Tanner, M. and Willingh, U. von (2014). Factors leading to the success and failure of Agile projects implemented in tradtionally waterfall environments. In: *Management, Knowledge and Learning International Conference - MakeLearn 2014*. Portorož, Slovenia, 693–701.

Tarhan, A. and Yilmaz, S. G. (2014). Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process. *Information and Software Technology* 56(5), 477–494. ISSN: 09505849.

Tseng, Y. and Lin, C. (2011). Enhancing enterprise agility by deploying agile drivers, capabilities and providers. *Information Sciences* 181(17), 3693–3708. ISSN: 00200255.

Vaidya, A. (2014). Does DAD Know Best, Is it Better to do LeSS or Just be SAFe? Adapting Scaling Agile Practices into the Enterprise. *Thirty-Second Annual Pacific Northwest Software Quality Conference*, 1–18.

Vijayasarathy, L. and Turk, D. (2012). Drivers of agile software development use: Dialectic interplay between benefits and hindrances. *Information and Software Technology* 54(2), 137–148. ISSN: 09505849.

Virmani, M. (2015). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. *5th International Conference on Innovative Computing Technology, INTECH 2015* (Intech), 78–82.

Walls, M. (2013). Building a DevOps Culture. *Velocity Web Performance and Operations*.

Webster, J. and Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly* 26(2), xiii–xxiii. ISSN: 02767783.

Weerd, I. van de and Brinkkemper, S. (2008). Meta-Modeling for Situational Analysis and Design Methods. *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*.

Wendler, R. (2013). The Structure of Agility from Different Perspectives. *2013 Federated Conference on Computer Science and Information Systems*, 1177–1184. ISSN: 2325-0348.

Wettinger, J., Breitenbücher, U., and Leymann, F. (2014). DevOpSlang - Bridging the gap between development and operations. In: *3rd Service-Oriented and Cloud Computing (ESOCC)*. Ed. by M. Villari, W. Zimmerman, and K.-K. Lau. Vol. 8745 LNCS. Manchester, UK: Springer, 108–122. ISBN: 9783662448786.

Wickramasinghe, V. and Widyaratne, R. (2012). Effects of interpersonal trust, team leader support, rewards, and knowledge sharing mechanisms on knowledge sharing in project teams. *VINE* 42(2), 214–236. ISSN: 0305-5728.

Wieringa, R. (2010). *Design science methodology*. Vol. 2, 493. ISBN: 9781605587196.

Wouter J.W. Geurts (2001). Faster, better, and cheaper. *IEEE Software* 18(3), 96–97. ISSN: 07407459.

Yin, R. K. (2011). Case study research: design and methods. *Evaluation & Research in Education* 24(3), 221–222. ISSN: 0950-0790.

Zahra, K., Azam, F., Ilyas, F., Faisal, H., Ambreen, N., and Gondal, N. (2017). Success Factors of Organizational Change in Software Process Improvement: A Systematic Literature Review. In: *5th International Conference on Information and Education Technology*. ACM, 155–160.

Zhu, L., Bass, L., and Champlin-Scharff, G. (2016). DevOps and Its Practices. *IEEE Software* 33(3), 32–34. ISSN: 07407459.

A Interview Protocol - Problem Investigation

Comparative Study on Agile DevOps Maturity & Team Performance at Financial Organizations
Interview Protocol
Department of Information and Computing Science

Utrecht University
Utrecht, The Netherlands

Introduction

I am Anitha Devi Nagarajan, a masters student following Business Informatics program at the Utrecht University. With regards to my graduation thesis that focuses on financial organization's Agile way of working specifically including DevOps, I would like to conduct an interview with you to collect data from the view of practitioners. Through my thesis, I am planning to bring up a competence model representing the different perspectives and factors that are necessary for the success of Agile and DevOps way of financial organizations' working. Based on that, I intend to develop a maturity model, which can be used as a maturity measurement tool. Moreover, I am planning to measure the performance metrics of certain organizations and compare it with their maturity level to see whether there is any correlation between them. I believe that the prospective maturity model will help the financial organizations not only to measure their current maturity level but also to improve their own performance metrics based on the maturity level from my assessment.

The interview is expected to take between 1 and 2 hours of time. During the interview, the questions will be asked to gain an insight on the following: drivers and goals for Agile and DevOps transitions, Agile & DevOps best practices from the perspectives of people, process, product, and organization, current and future Agile & DevOps practices of interviewee's organization from the perspectives of people, process, product, and organization, and Performance metrics related to the DevOps implementation goals.

The interview data will help to me to gain the data from practical point-of-view as well as to narrow down my research data on financial organizations. However the

information collected from the interviews will be used together with the details collected from the literature in order to develop the research artefacts (i.e. Competence model and Maturity model).

Confidentiality

During the interview, I will be recording the complete interview for transcribing purpose. All the information gathered during the interview will be treated with respect and will be used only for scientific purposes. The interview data will be held confidential and it will not be shared anywhere outside the university. The audio recordings will be deleted at the end of the research. The effort will be made to preserve the interviewee's confidentiality by following code assignments for participant's name and organization.

Voluntary Participation

Your participation in this study is voluntary. It is up to you to decide whether or not to take part in this study. If you decide to take part in this study, you will be asked to sign this consent form before the interview begins. After you sign the consent form, you are still free to withdraw at any time and without giving a reason. Withdrawing from this study will not affect the relationship you have, if any, with the researcher. If you withdraw from the study before data collection is completed, your data will be returned to you or destroyed.

Benefits

Once the research is completed, the research report which consists of the research artefacts as well as the comparative study results will be sent to you. The competence model will depict all the aspects that any financial organization will require to embrace DevOps along with Agile practices and so it can be used as a reference for the DevOps implementation in the Agile environment. The maturity model will be of help to assess and move forward with the maturity level. Besides these, the analysis on the relationship between performance metrics and the maturity level may disclose some interesting facts between them. Thus, your organization can take advantage of all the benefits by utilizing the artefacts and be informed of the comparison study results.

Consent

I have read and I understand the provided information and have had the opportunity to ask questions. I voluntarily agree to take part in this study.

Participant 's signature _____ Date _____

Researcher 's signature _____ Date _____

First of all, I would like to thank you for accepting to participate in this interview. As I have mentioned in the informed consent document, I am going to ask questions about your organizational practices regarding those factors that are involved in the Agile adoption and DevOps implementation strategy of your organization.

General Questions

Firstly, I would like to ask some general questions about you and your organization.

1. Can you please elaborate your role and general responsibilities within your company?
2. Can you please let me know when did your team(s) transform to Agile and (then to) DevOps?
3. Can you please give me an overview of roles present in your AgileDevOps organization? (number of teams, number & name of roles present within each team, other common roles across teams etc.)
4. Are you following any specific Agile methods such as Scrum and XP within your department? If so, what is it? How long have you been following this Agile method?

Agile Questions

Secondly, I would like to ask questions about Agile adoption factors. In order to achieve success with Agile adoption, there are several factors required to be kept in place by organizations. Various studies have revealed that these factors belong to dimensions such as organizational, people, process and technical dimensions (Darwish and Rizk, 2015; Hameed et al., 2016).

Organizational dimension of Agile adoption

In the case of an organizational dimension, it was found that a cooperative corporate culture and a collaborative organizational environment are the most influencing factors of Agile success (Chow and Cao, 2008; Hameed, Latif and Kholief, 2016).

1. Do you think that Agile has been universally accepted and followed within your organization?
   (a) If yes, can you please justify your answer with some supporting statements?
   (b) If not, where do you think that the acceptance is lacking?
2. Can you please explain how does the management and executive level people influence the success of your Agile projects?
3. Other than normal day-to-day development work, do you recognize that Agile has impacted any other organizational activities such as team distribution and internal reward system?

4. Has Agile brought any changes into the corporate culture of your organization?
   (a) If yes, can you please point out such changes?
   (b) If not, do you think that the current corporate culture is supporting agile? How?
5. What are all the future changes you expect to be brought into this dimension to sustain agility within your organization?

People dimension of Agile adoption

Next, I would like to move on to people dimension of Agile adoption. Within this aspect, literature identifies customer involvement and team capability as the main success factors regarding Agile (Darwish and Rizk, 2015; Hameed, Latif and Kholief, 2016). Moreover, Agile values customer collaboration over contract negotiation (Value 3).

1. How is the customer involvement is ensured throughout the Agile software development lifecycle within your organization?
2. How does your organization help to keep up the bonding between team and customers?
3. What are all the future practices to be implemented within your organization in order to improve and maintain the customer relationship?

Process dimension of Agile adoption

Understanding organizational and people aspects, I want to switch over to the process dimension of Agile adoption. Previous studies have identified that it is imperative to have a good project management process and project definition process for attaining success with Agile projects (Darwish and Rizk, 2015; Hameed, Latif and Kholief, 2016). Although agile encourages to welcome changes (principle 2), ill-defined project requirements would definitely impair the success of any project (Chow and Cao, 2008).

1. Can you please elaborate agile project requirements elicitation practices that are being followed (along with roles involved) within your organization?
2. How do you manage with the changes being introduced during the development process?
3. What are the future practices to be implemented in order to improve project scoping and requirements definition processes within your organization?

Agile emphasizes on continuous improvement through learning from mistakes (principle 12) and supporting Agile methods like Scrum, support this principle through various time-boxed events such as daily and retrospective meetings.

1. Could you detail on the activities that control the status of the project within your organization?
2. How do these activities influence the day-to-day work of team members?

3. How does your organization help your team to participate in these activities with motivation?
4. What are the future improvements you anticipate within your organization in order to improve such Agile progress tracking activities?

Technical dimension of agile adoption

Last but not least, technical factors play an important role in any Agile project success by selecting appropriate Agile method, practices, tools and techniques (Darwish and Rizk, 2015; Hameed, Latif and Kholief, 2016).

1. What are the all the practices being followed within an Agile project development process within your organization?
2. How are these practices supported with related tools?
3. What are the future improvements to be made in terms of practices and tools within your organization's Agile software development projects?

Before I move on to DevOps, I would like to ask the final question regarding Agile Implementation within your organization:

1. Can you please explain how do you measure your defined objectives regarding Agile implementation within your organization?

DevOps Questions

DevOps drivers and general challenges in DevOps adoption

Lastly, I would like to ask questions about DevOps implementation factors. From the literature, it is understood that DevOps practices have evolved through cloud-based product organizations such as Facebook, Flickr and Netflix (Zhu, Bass, and Champlin-Scharff, 2016). These organizations are cloud based product companies, whereas financial organizations such as banks are utilising various old and new technologies to develop multiple interconnected systems.

1. In this case, can you please elaborate why banks are interested in following DevOps?

Agile software development methodology has already proven to bring improvements such as faster software development with enhanced quality and the ability to welcome changes throughout the project, over traditional software development methods such as waterfall (Bhadoriya, Mishram, and Malviya, 2014) and iterative development (Tarhan & Yilmaz, 2013).

1. What are the drivers for your organization to implement DevOps while you were already following Agile principles and practices?
2. How do you measure the achievement of these goals related to your organization 's DevOps implementation?

There are several challenges reported by organizations in their path towards success DevOps implementations such as lack of support by management and/or team, technical challenges, challenges with adapting organizational process in support of DevOps (Hamunen, 2016).

1. What are such difficulties that banks face with DevOps implementations?
2. How are these difficulties being tackled within your organization?
3. Considering the current situation, how do you think that these difficulties will be overcome in the future?

Factors involved in DevOps implementation

Next, I want to know more about the factors that require changes due to DevOps implementation. In order to understand the details better, I would like to ask questions about the factors from the discussed agile factor dimensions of organization, people, technical and process.

Organizational dimension of DevOps adoption

Previous studies have identified that the DevOps culture stipulates organizations to practice aligned incentives and responsibilities among various roles participating in the project, rewarding individuals for their contribution towards teamwork without just focusing on individual performance, continuous and frequent feedback that motivates to improve, but not blaming the mistakes of team members (Walls, 2013).

1. What are all the changes that you observed in your organizational culture since DevOps has been introduced?
2. Which are all the practices that your organization has tried/is trying to implement in order to bring in the DevOps mindset within your organization?
3. What are all the practices that your organization is planning to implement in order to enhance the DevOps culture within your organization?

People dimension of DevOps adoption

Further, I would like to discuss about the people aspect within DevOps. DevOps advocates collaborative teams, whose members have shared goals, growth mindset that lead to blameless culture (Walls, 2013).

1. What do you think are the essential characteristics of a DevOps team member to be part of such a collaborative DevOps team?
2. How does your organization help and motivate people to get the right DevOps mindset and so the required characteristics?
3. What practices have been fostered to grassroot the collaborative culture within the team(s)?
4. What are all the future actions that are planned to be implemented to improvise the people factor within your DevOps organization?

Process and technical dimension of DevOps adoption

DevOps facilitates process automation for various advantages regarding speed, quality and harmony of processes (Swartout, 2014). Furthermore, automation will also help people to share their responsibilities and thus it also aids to collaborative culture.

1. Can you elaborate how significant is automation within your organization's DevOps work culture?
2. Which are all the processes that have already been automated within the Agile software development and delivery of your organization?
3. What are all the future automation practices that are planned to be implemented further?
4. Is your organization facing any challenges regarding the process automation (due to the nature of your application or technology)?
5. What actions are taken to mitigate these challenges?

The increased process automation enables the usage of continuous software engineering practices such as Continuous Integration, Continuous Delivery and Continuous Deployment (Davis and Daniels, 2015). Continuous Integration is the frequent integration of newly written code into actual software's existing code (Davis and Daniels, 2015). Continuous Delivery is the ability to deploy software at will (Neely and Stolt, 2013). Continuous Deployment is the frequent deployment of incremental software update into production (Savor et al., 2016). These continuous software engineering practices include all the activities from planning till maintenance including monitoring and innovation.

1. Which of such continuous software engineering practices are being exercised within your organization?
2. Which of such continuous software engineering practices are planned to be utilized in the future?
3. Do you have any challenges in embracing any of these continuous software engineering practices within any of the Agile software development and delivery (and maintenance) activities?

B Appendix - ICoding Scheme

The following table presents the codes that refers to the interviews performed with experts regarding the problem investigation. To keep the confidentiality, we replace the interviewee's names with their job position.

Code	Purpose	APA Reference
iv-pi-1	Problem investigation interview	(Scrum Master, personal communication, January 24, 2018)
iv-pi-2	Problem investigation interview	(Project Manager, personal communication, January 8, 2018)
iv-pi-3	Problem investigation interview	(DevOps team member, personal communication, January 10, 2018)
iv-pi-4	Problem investigation interview	(DevOps team member, personal communication, January 10, 2018)
iv-pi-5	Problem investigation interview	(DevOps team member, personal communication, January 20, 2018)
iv-pi-6	Problem investigation interview	(Agile Coach, personal communication, January 22, 2018)
iv-pi-7	Problem investigation interview	(DevOps team member, personal communication, November 22, 2017)

C Interview Protocol - Artifacts Validation

DevOps implementation framework for Agile based large financial organizations

Interview Protocol

Department of Information and Computing Science

Utrecht University
Utrecht, The Netherlands

Introduction

I am Anitha Devi Nagarajan, a masters student following Business Informatics program at the Utrecht University. With regards to my graduation thesis, I have developed a DevOps implementation framework based on the data collected from practitioners working at Agile based financial organizations and the data gathered from the scientific literature. Today I am going to present the following main artifacts that were resulted from this study.

- Drivers for DevOps implementation besides Agile implementation at large financial organizations (artifact 1)
- DevOps implementation framework developed with the main perspectives and focus areas identified from this study (artifact 2)
- The relationship between the identified drivers and the developed DevOps implementation framework (artifact 3)

The total session is expected to last between 45 and 60 minutes. During the session, I will be running an interactive presentation in which I will be presenting the artifacts and then asking questions regarding them. The questions can be found at the next page of this document and they are formulated as such to evaluate the artifacts based on several criteria namely, **completeness, correctness, usefulness, understandability, and fit with the organization**.

Your participation in this session is voluntary and so if you do not feel comfortable or do not wish to answer any of the questions, you are ever free to do so. Your answers will be recorded but the recordings will be deleted after utilizing the collected data, and will never be disclosed to any third parties.

Consent

I have read and I understand the provided information and have had the opportunity to ask questions. I voluntarily agree to take part in this study.

Participant 's signature _____ Date _____

Researcher 's signature _____ Date _____

Questionnaire

Interviewee Profile

Following questions are to understand your profile and your role in a DevOps implementation.

1. Can you please elaborate your role and general responsibilities within your company?
2. Can you please let me know when did your team(s) transform to Agile and (then to) DevOps?
3. How far do you think that your department or team has been into DevOps way of working?

Artifacts validation against criteria

Please answer the following questions with a 'Yes' or 'No' but kindly provide your short reason when the answer is 'No'.

Artifact 1: Drivers for DevOps implementation

Criteria: Completeness & Fit with organization

1. Do you think that this list covers your organizational drivers?
2. Do you think that there are any other important drivers for your organization to go for DevOps implementation?

Criterion: Understandability

1. Are the given names of drivers sound straightforward to understand?

Artifact 2: DevOps implementation Framework inclusive of perspectives and focus areas

Criterion: Completeness

1. Do you think that the different perspectives include all the different dimensions involved with the DevOps implementation?
2. Do you agree with the identified focus areas within the perspective [1,2,3,4]?
3. Does any of the focus areas seem unnecessary within the perspective [1,2,3,4]?
4. Do you think any other important focus area is missing from the given list of perspective [1,2,3,4]?

Criterion: Understandability

1. Is the framework clear and easy to understand?
2. Are the terms chosen for the perspectives and focus areas sound coherent and clear?

Criterion: Fit with organization

1. Is this framework suitable to be used within your organizational environment?

Criterion: Usefulness

1. Do you think that this framework is useful for you?
2. Do you think that this framework will help with the ongoing DevOps implementations and/or future DevOps implementations within your organization?
3. Do you think that this framework is suitable to bring awareness about DevOps implementation at different levels of employees such as managers and team members?

Artifact 3: Relationship between Drivers and the framework

Criterion: Accuracy

1. Do you agree with the relationship between driver[1...6] and its corresponding framework variation? If not, which specific focus area/perspective and driver combination that you have concerns with?

Criterion: Usefulness

1. Do you think that these framework variations can help to keep your DevOps implementation focus aligned with your goals?

Open questions

1. What is your overall opinion about this framework and its variations corresponding to drivers?
2. Whom do you think are the best target audience for this framework?
3. Would you recommend using this framework as a guideline for your colleagues, why?
4. Do you have any other comments?