



Master Thesis

DevOps Implementation Model
for Large IT Service
Organizations

Margot Jonker
January 2017

This page intentionally left blank

DevOps Implementation Model for Large IT Service Organizations

Master thesis submitted to Delft University of Technology
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in Systems Engineering, Policy Analysis & Management

Faculty of Technology, Policy and Management

by

Margot Jonker

Student number: 4397975

To be defended in public on January 19th, 2017

Graduation committee

Chairperson:	: Prof.dr.ir. M.F.W.H.A. Janssen, Section ICT
First Supervisor	: Mw. drs. J. Ubacht, Section ICT
Second Supervisor	: Dr. M.E. Warnier, Section SES
External Supervisor	: S. Frijling, MSc., Accenture, Department Infrastructure Consulting



Preface

This report describes my thesis project on a DevOps implementation model for large IT service organizations. The project is done for the Master's degree of Systems Engineering, Policy Analysis & Management with the additional track Information Architecture at the Delft University of Technology. The project is done in combination with an internship at Accenture, which facilitated the research process.

In this preface, I would like to thank my graduation committee. First of all, many thanks to Jolien, who guided me through the whole process as my first supervisor. She helped me to keep on going in the right direction when I was sometimes too focused on unnecessary details. Next to that, she made always time for me which made the collaboration very nice. Also, I would like to thank Martijn as second supervisor. His feedback was very helpful as it was from another perspective. Next, I would like to thank Marijn for being the chairman of my graduation committee. He always provided critical, but relevant feedback on the moments it was necessary.

Finally, I would like to thank my supervisor of Accenture. Sebastiaan, thank you for supporting me during my whole internship. You were always open for questions and helped me where possible by thinking in opportunities. You introduced me to many people and let me experience the way of working at Accenture. Due to your help and Accenture's collaboration, I was able to retrieve a lot of relevant information for the research.

Margot Jonker

Delft, January 2017



Executive Summary

Currently, our society is in the middle of the digital revolution; digital devices are more and more integrated in our society. In a rapidly changing business environment, it is important for organizations to keep adding value to customers. Being flexible and being able to innovate at a fast pace is key nowadays for IT service organizations to stay competitive. Therefore, IT service organizations need to decrease the time to market within their current business model. Ideally, the time to market should be continuous, which means to say that service delivery is continuous. Therefore, organizations aim to transform their business processes from incremental processes into Agile processes. However, many of those practices are focused on aligning the business and development phases with iterative processes. The operational department is still not aligned with this approach. An answer to this shortcoming is DevOps.

DevOps extends the Agile way of working to the operational department within an organization. The definition that is created and used within this research is: *“DevOps is a cultural movement that breaks silos between the business, development and operations department, combined with a number of service development practices regarding people, process and technology, that enable rapid development and delivery of services”*. DevOps extends the Agile with its values of systems thinking, focus on flow, amplify feedback loops and culture of continual experimentation & learning. These values are translated to many practices related to people, process and technology. As these practices are integrated within the whole organization, there are also many challenges regarding people, process and technology. Small IT organizations, that just entered the market, can easily change or upgrade their relative new way of working as they can build their organization from scratch. Large organizations have more difficulties as they have to cope with all kind of legacy systems regarding people, process and technology. Therefore, most large organizations face the challenge of how they have to tackle the implementation of DevOps. The research aims to develop a DevOps implementation model that can help IT managers of large IT service organizations which have adopted an Agile way of working with their process of implementing DevOps. The perspective of an IT manager is taken to investigate the implementation of DevOps. This model will indicate the most important aspects for a IT manager to focus on for implementing DevOps within a large IT service organization.

The research is constructed according to the design science research framework of Peffers et al. (2007). An adapted version of the framework is used to perform a suitable design science research. The problem identification & motivation phase introduces the research problem, research objective and research scope. This extended with a description of IT service organizations and concepts related to DevOps based on literature research and exploratory interviews. Based on the definition of DevOps, the phase objectives of solution is indicated with DevOps challenges related to people, process and technology. After that, the Design & Development phase is started and contains three stages. First, a theoretical conceptual model for DevOps is designed based on literature research. The value of this model for large IT service organizations is assessed by conducting case interviews at such organizations and an expert evaluation. This is the second stage of the development phase. Based on the theoretical and empirical conclusions, a final implementation model is developed in the third and last stage of the Design & Development phase. Finally, the Demonstration phase is used to illustrate the use of the model with providing short use cases.

Within the research, the implementation of DevOps is approached as a change in the business process of an organization. Business process change models are therefore used to develop a theoretical conceptual model for the research. Based on literature research, a combination of a business process reengineering framework and the elements of lean six sigma are chosen as in that way both top-down and bottom-up approaches are included. The categories of this model are business process, people, technology, information, organizational culture and structures in which multiple DevOps aspects are categorized. The conceptual model is a general model for a change towards DevOps based on literature.

To make the model specific for large IT service organizations, empirical research has been done at such organizations. Nine cases are described by conducting case interviews with IT managers of large IT

service organizations. The empirical data consist of insights on the current implementation and focus of organizations towards DevOps aspects. First of all, it is confirmed that there is no consensus in defining DevOps. Organizations and people are all defining DevOps differently, although the starting point of research that states that DevOps is a cultural movement is evaluated as correct. After concluding this, the research definition of DevOps is used during the whole research to create the same starting point. The empirical results show per category an indication of aspects that are relevant for DevOps, aspects that would be nice to have, aspects that already implemented due to an Agile way of working and aspects that are not DevOps specific within large IT service organizations. The results also indicate a certain ranking in which the aspects can be placed although this order can differ per organization. Next to the case interview results on the model, additional findings during those cases related to DevOps aspects and related to the implementation are summarized. Most additional findings related to the implementation process are evaluated by an expert evaluation session in order to analyze its correctness. The conclusions of the case interviews together with the conclusions of the expert evaluation session provide a clear overview of all important aspects that are relevant for large IT service organizations regarding an implementation of a DevOps way of working.

Based on those conclusions, a final implementation model is developed. Although all components are indicated as relevant by the research, it was concluded that the aspects and the implementation approach of the DevOps way of working are both dynamic. All components and aspects can fluctuate in level during the implementation. Next to that, the organizational factors are limiting or adjusting the way in which DevOps can be implemented. Also the implementation itself is dynamic and requires an adaptive process and project approach. Therefore, the building blocks of the model are: external drivers, components with its aspects, the implementation approach and critical organizational factors. Composing these building blocks retrieves a final model for DevOps implementation at large IT service organizations that is proposed by the researcher. The model is presented in a simplified visualization in which the most important factors are made visible with an associated table to provide more details on the relevant aspects of the components to focus the implementation on.

The model supports the implementation of DevOps at a large IT service organization that is working Agile. From the point of view of an IT manager, the focus areas and common aspects of DevOps that are relevant are indicated. This can help the IT manager to indicate the focus points of his DevOps implementation. This choice is impacted by multiple organizational factors that are specifically present within each organization. Based on these organizational factors, the best focus for the DevOps implementation can be determined. The model also presents relevant aspects on the specific implementation approach. The use of the model is illustrated by describing multiple organizational situations in which the model can help to indicate the relevant focus areas and aspects. The research also provides an extra concept tool for performing an assessment on the current DevOps situation at an organization.

It is recommended to extend the assessment tool and the implementation model with more operational actions on the next steps to be done. Future research on this topic could be an in-depth case study or survey to validate the model or performing a new research by taking a system dynamic approach to the implementation of DevOps as it operates in a complex and dynamic area.

Table of Contents

Preface.....	4
Executive Summary	6
List of Figures	10
List of Tables.....	11
List of Abbreviations.....	12
1. Introduction	14
1.1. Chapter Introduction.....	14
1.2. Situation.....	14
1.3. Research Problem.....	15
1.4. Research Objectives	15
1.5. Research Methodology.....	17
1.6. Chapter Conclusion	19
2. DevOps & Related Concepts.....	22
2.1. Chapter Introduction.....	22
2.2. Domain Description	22
2.3. DevOps.....	30
2.4. Chapter Conclusion	36
3. Design of the Theoretical Model.....	39
3.1. Chapter Introduction.....	39
3.2. Process Change Model	39
3.3. Theoretical Model for DevOps.....	44
3.4. Chapter Conclusion	48
4. Development of the Implementation Model.....	51
4.1. Chapter Introduction.....	51
4.2. Research Method.....	51
4.3. Case & Expert Selection.....	51
4.4. Case Interview Protocol	53
4.5. Results	55
4.6. Evaluation of the Additional Findings	76
4.7. Chapter Conclusion	81
5. Development of the Final Implementation Model	85
5.1. Chapter Introduction.....	85
5.2. Design of Final Model.....	85
5.3. Building blocks.....	86
5.4. Presentation of Final Implementation Model	91
5.5. Assessment Tool.....	94
5.6. Demonstration	95
5.7. Chapter Conclusion	97

6. Conclusion.....	100
6.1. Chapter Introduction.....	100
6.2. Conclusions	100
6.3. Limitations.....	107
6.4. Reflection	108
6.5. Future Research.....	110
6.6. Recommendations	111
References	113
Appendices	119
Appendix A – Overview of Exploratory Expert Interviews.....	119
Appendix B – Interview Guide Experts	120
Appendix C – Interview Guide Case Interviews	121
Appendix D – Questions Evaluation Session.....	125
Appendix E – Case Descriptions.....	126
Appendix F – Results Detailed Tables	130
Appendix G – Results Evaluation Session	135
Appendix H – Implementation Assessment Tool.....	138
Appendix I – Scientific Article.....	139

List of Figures

Figure 1: The Research Flow Diagram	20
Figure 2: The Strategic Alignment Model of Henderson & Venkatraman (Maes, 1999)	23
Figure 3: Example of processes between departments (Bon, 2004).....	24
Figure 4: Position of SLAs and OLAs in ITSM.....	24
Figure 5: A high-level delivery pipeline of an organization	25
Figure 6: Software development lifecycle within the waterfall method.....	26
Figure 7: Basic principles of the Agile Manifesto (Beck, 2001).....	27
Figure 8: Indication of DevOps within delivery pipeline.....	28
Figure 9: Overview of the walls of confusion	29
Figure 10: Spotify Model (Kniberg & Ivarsson, 2012)	29
Figure 11: The Three Ways of DevOps (Kim, 2013).....	32
Figure 12: The software delivery pipeline with indication of different concepts.....	34
Figure 13: Overview of all BPC concepts (Kristekova & Jurisch, 2012)	40
Figure 14: BPR framework (Kettinger & Grover, 1995)	41
Figure 15: Lean Six Sigma (Cucoranu, Parwani, & Pantanowitz, 2014)	42
Figure 16: Theoretical design of model.....	43
Figure 17: Overview of fixed and variable measurements within Waterfall and Agile	44
Figure 18: Overview of relevant DevOps aspects within the categories of the theoretical model	49
Figure 19: Overview of the Focus Areas.....	57
Figure 20: Importance of the focus areas	58
Figure 21: The degree of the focus areas.....	59
Figure 22: Graph with overview of answers on environmental factors.....	59
Figure 23: Graph with overview of all answers on Performance Measurements	60
Figure 24: Graph with all deltas for the aspects within the category "Process"	62
Figure 25: Overview of the conclusions within the category "Process"	63
Figure 26: Graph with all deltas for the aspects within the category "People"	64
Figure 27: Overview of conclusions within the category "People"	65
Figure 28: Graph with all deltas for the aspects within the category "Information"	65
Figure 29: Overview of the conclusions within the category "Information"	66
Figure 30: Graph with all deltas for the aspects within the category "Org. Culture & Structure"	66
Figure 31: Overview of the conclusions within the category "Org. Culture & Structure"	67
Figure 32: Graph with all deltas for the aspects within the category "Technology"	67
Figure 33: Overview of the conclusions within the category "Technology"	68
Figure 34: Graph with the overview of all deltas for the category "Change management"	69
Figure 35: Overview of the conclusions within the category "Change management"	70
Figure 36: Legend associated to Table 5: Overview of all aspects per category	71
Figure 37: Overview of all additional findings	75
Figure 38: Overview of the answers on the first question, visualized in a word cloud.....	76
Figure 39: Overview of the answers to the question on the focus areas of DevOps	77
Figure 40: Legend for Table 10 and Table 11	82
Figure 41: Overview of the components	87
Figure 42: Legend of Table 13	88
Figure 43: Example of different implementation maturities per component	88
Figure 44: Building block of implementation approach.....	89
Figure 45: Legend associated with Table 16.....	93
Figure 46: Simplified visualization of implementation model	94
Figure 47: Overview of the visualized model for the use cases	95
Figure 48: Legend for Table 17.....	104
Figure 49: Implementation model (visualization)	105

List of Tables

Table 1: Overview of DevOps definitions found in literature.....	30
Table 2: Differences of waterfall, Agile and DevOps	33
Table 3: Overview of the characteristics of the cases	52
Table 4: Example of the interview guide per category.....	54
Table 5: Overview of all aspects per category ranked according to empirical data	70
Table 6: Example Statements	77
Table 7: Overview of all statements and the given answers.....	78
Table 8: Overview of the evaluated aspects by the expert evaluation session	80
Table 9: Overview of evaluated aspects related to the implementation	81
Table 10: Overview of the conclusions upon the implementation	82
Table 11: Overview of conclusions upon the categories within model.....	83
Table 12: Overview of external factors	86
Table 13: Overview of the components and its aspects.....	87
Table 14: Overview of conclusions on implementation approach	89
Table 15: Overview of organizational factors	91
Table 16: The relevant focus areas and aspects of DevOps implementation	93
Table 17: Implementation model (table).....	104
Table 18: Overview of implementation steps of BPC models (Ketinger, Teng, & Guha, 1997; Nave, 2002).....	106

List of Abbreviations

BPC	Business Process Change
BPR	Business Process Reengineering
CD	Continuous Delivery
CI	Continuous Integration
COBIT	Control Objectives for Information and Related Technology
DAD	Disciplined Agile Delivery
DevOps	Development & Operations
DSR	Design Science Research
IaaS	Infrastructure as a Service
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
KPI	Key Performance Indicator
LeSS	Large Scale Scrum
LSS	Lean Six Sigma
MVP	Minimal Valuable Product
PaaS	Platform as a Service
QA	Quality Assurance
SaaS	Software as a Service
SAFe	Scaled Agile Framework
SDLC	Software Development Lifecycle
TOGAF	The Open Group Architecture Framework
TQM	Total Quality Management
VoC	Voice of the Customer
WoW	Way of Working
XaaS	Everything as a Service

Chapter 1

Introduction

1. Introduction

1.1. Chapter Introduction

This chapter provides an overview of the research problem, research objective and research approach. The introduction helps to understand the purpose of the research and the methods used within the research. First, the situation in the current world related to DevOps is explained in section 1.2. Section 1.3 dives in the complication within the situation and formulates the research problem. Based on the problem, the research objectives and the main research question are presented in section 1.4. Next, the general research methodology is explained in section 1.5 and section 1.6 concludes the introduction and summarizes the chapter structure to guide the reader through the report.

1.2. Situation

Currently, our society is in the middle of the digital revolution; digital devices are more and more integrated in our society. The business environment is rapidly changing and according to the Accenture Technology Vision 2016, it is even expected that 25% of the world's economy will be digitalized by 2020 (Nanterme & Daugherty, 2016). In a rapidly changing business environment, it is important for organizations to keep adding value to customers by being innovative. According to Salehi & Yaghtin (2015), "innovation could be recognized as a key success factor in an increasingly competitive, global economy" (Abstract). Many organizations within different sectors are thereby changing their business strategy, often resulting in offering Information Technology (IT) services. These IT services can be the new main business of an organization or as a supportive service of the main business. Digitalization is a global trend and multiple digital concepts are finding their way to the business field, such as digital workspaces, digital government platforms and Internet of Things (Howard, 2015). Due to the ongoing emerge of new technologies, customer's expectations about the flexibility and speed of IT services are increasing (Emidio et al., 2015). Being flexible and being able to innovate at a fast pace is key nowadays for IT service organizations to stay competitive (Colavita, 2016). Therefore, IT service organizations need to decrease the time to market within their current business model. Ideally, the time to market should be continuous, which means to say that service delivery is continuous. Continuous delivery of services implies a continuous delivery of software.

When focusing on software engineering, organizations have adopted all kind of process optimizations in their software development practices (Virmani, 2015). Traditionally software development was done according to the waterfall method with a stepwise process, where nowadays organizations are switching to a more Agile development (Barlow et al., 2011). Conboy (2009) defined Agile as "the continual readiness of an information systems development method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment." (p. 340). However, while focusing on an Agile development, optimizing the operational phase is not taken into account. Software development within organizations is done by the development department and another department is handling the implementation and maintenance of the software, often called the operations department. Two different teams are responsible for their own software activities within the related phase: development and operations. The process of both departments are often not aligned with each other as Virmani (2015) states: "software development teams are able to deliver at a much faster pace than the pace at which operations teams can absorb the builds." (p. 78) The built software of services is managed by operations teams on site and those teams are used to working in a different manner than development departments. Their mindset and goal is also different; development is focused on changes and innovation, where the operations department is focused on reliability. This barrier refers to the existence of so-called silos; separate business units which are not aligned with each other, for example in terms of organizational structure or available information (Serrat, 2010). Such differences can also be seen between the different layers of services, such as applications and infrastructure, both necessary for providing IT services. Teams associated to applications are more appropriate to adopt an Agile way of working, due to the nature of activities. Traditional IT infrastructure must be stable and is hardly flexible, where applications are more open for

changes. Due to these silos, ‘Agile’ organizations are not able to provide the optimal time to market, which ideally is continuous.

To remove the barriers between the silos, a relatively new concept can be applied, called DevOps. Although different definitions of DevOps are known, DevOps can be seen as a movement, including multiple practices, which aims for creating more flexibility and innovation throughout the whole business process of creating an IT service. DevOps extends the Agile way of working in a way of using Agile practices to increase the collaboration between the development and the operations staff (Colavita, 2016). Colavita (2016) states that “by using Agile/lean techniques, it allows for IT services to be updated continuously so the business can capture market opportunities and reduce time for addressing customer needs.” (p. 203) Within DevOps, operations will become a valued member of the traditional Agile process with equal rights (Debois, 2011). Adopting a DevOps way of working seems to be beneficial as it promises to result in cost reductions, high throughput of innovations, fast time to market and improvement of quality (Kim, 2011). These benefits are similar to the Agile benefits, though people behind DevOps state that DevOps is increasing the proposed benefits even further (Interview 2).

1.3. Research Problem

Implementing DevOps seems to have significant benefits over other approaches, only it is not just simply implementing a tool or technology. The implementation of the practices within DevOps is touching multiple processes, structures and technology. However, organizations cannot easily change entire processes and their way of working. To successfully implement such new concept, current structures and processes have to be analyzed carefully. It will take multiple iterations and evaluations to reach for a DevOps way of working and still it is believed the perfect DevOps way of working does not exist yet (Debois, 2011). Within an organization many legacy barriers exist for implementing a new way of working. Small IT organizations, that just entered the market, can easily change or upgrade their relative new way of working as they do not have much fixed establishment yet. An example for this is the company Spotify with their Spotify business model. They used an Agile method “Scrum”, “which consists of practices to improve project management by quickly exposing risks within the project”, following Yu & Petter (2014, p. 912). Spotify experienced that this method was lacking in its performance for their growing amount of teams and therefore they developed an own Spotify engineering culture throughout the whole business process, related to DevOps, which include many aspects of an Agile way of working (Kniberg, 2014).

Large organizations offering all kind of IT services, where this research project is focusing on, have more difficulties regarding the implementation of DevOps as they have to cope with their legacy systems (Boehm & Turner, 2005). Barriers due to legacy systems can be found in different aspects within an organization, for example in currently used information systems, roles, structures or the current way of working. Accenture, the IT consultancy company that is involved within the research, has multiple large IT service organizations as clients. Accenture notices that those clients are wondering which benefits DevOps can deliver for their organization and what the implementation strategy must be (Interview 2). Such clients have often implemented an Agile way of working already and aim for continuous delivery throughout the whole process to survive within the rapidly changing business environment. However, they are facing the big challenge of changing their organizational processes and structures with implementing DevOps. Organizations and experts in this field are searching for best practices regarding DevOps, although opinions still vary a lot on what DevOps includes. Consultants at Accenture are also experiencing this lack of proven in-depth knowledge and they desire more research insights on the implementation of DevOps to base their advice on. As DevOps originated from 2009, there is not much scientific literature on the topic available. Therefore, there is a need to study the implementation of DevOps within large IT service organizations that already have an Agile way of working.

1.4. Research Objectives

Based on the indicated research problem, the aim of the research is to create more insight on the implementation of a DevOps way of working at large IT service organizations in varying industries that already have an Agile way of working. In order to create this insight, a DevOps implementation model is developed within this research. For the development of this model, it is necessary to conduct research

on the focus points for the implementation of DevOps at such organizations. By focusing on the role of a IT manager, these focus points could support the process of adopting a DevOps way of working at large IT service organizations as these points can indicate on which aspects of DevOps IT managers of such organizations have to focus its implementation. Based on these focus points, a suitable implementation approach could be determined to support the implementation process of DevOps in a large organization offering different IT services.

Therefore, the main objective of the research is formulated as follows:

To develop a DevOps implementation model that can support IT managers of large IT service organizations, which have adopted an Agile way of working, with their process of implementing DevOps

1.4.1. Scope

The research focuses on large enterprise organizations in which large is defined by the number of employees, being preferably over 1000 employees. The organizations have to offer IT services, as main business or as a supportive process of the main business. The type of IT services that an organization offers is not relevant as the foundation of IT services contains comparable elements. IT service organizations within different sectors are included in the research as the objective is to draw common, cross-industry conclusions. Accenture has many large organizations as clients, which operate in varying sectors. Therefore, the cross-industry scope is useful from their perspective.

IT managers play a critical role in any DevOps transformation (PuppetLabs, 2015). The problem owner is therefore the IT manager of a large IT service organization. The IT manager can be responsible for both phases development and operations or only for one within the organizations. The objectives of an IT manager are according to Williams & Baxter (1996) “1) better aligning IT product and services with the firm’s strategic objectives, 2) delivering solutions faster, and 3) providing high-quality, cost effective support” (p. 32). A DevOps way of working can help the IT manager to realize these objectives. Therefore, it is useful to indicate the focus points of the implementation for this role and this helps also to limit the research from becoming ambiguous. Kristekova & Jurisch (2012) refer to the fact that “the higher the level of ambition, the larger the number of critical activities that need to be tackled and the more organizational aspects that have to be changed” (p. 459). To decrease the level of unmanageable ambition, all organizational aspects included within the research are viewed from the IT manager’s perspective. By focusing on the role of IT manager, it is possible to identify the crucial changes that are necessary for the implementation of DevOps.

Another delineation is made regarding the current way of working of an organization. Each organization has its own strategy with an underlining methodology for all business processes, thus also for the IT process. As DevOps can be seen as expanding the Agile methodology over multiple phases of the business process, the research focusses on organizations that are working already Agile. These organizations are most likely thinking of moving to DevOps or are already moving to a more DevOps way of working. Regarding the way of working for the IT service management, the ITIL framework has become popular within organizations (Pollard & Cater-Steel, 2014). ITIL contains some best practices found across the range of IT service providers and offers a framework for the delivery of quality of IT services (Bon, 2004). It can be assumed that most large organizations have already implemented ITIL in a certain maturity, however, the adoption of ITIL is not a strict requirement for the research.

1.4.2. Scientific Relevance

DevOps is a relative new term and there is not much scientific literature available yet. There is a lot of so-called grey literature on blogs and conferences that describe experiences on practical case studies. Grey literature means, according to McAuley et al. (2000), “studies that are unpublished, have limited distribution, and/or are not included in bibliographic retrieval system” (p. 43). As much of this literature is seen as grey literature, it can be used within the research, though with keeping this limitation in mind. Literature available is often on which practices are possibly covered within DevOps. This is often learned from the adoption to small organizations, not on large organizations. Most of the time it is also focused on the more technology aspect, continuous delivery, whereas the way of working with a new

culture and people aspects are discussed in less detail. It can be seen as a research gap that literature does not describe how the implementation of a DevOps way of working is or should be approached within large organizations. The research will focus on implementing DevOps in the current way of working in the business process of such organizations. This research can contribute to the limited literature on the implementation of DevOps by indicating the relevant focus points of DevOps at large IT service organizations.

1.4.3. Societal Relevance

From the societal perspective, there are two main parties that could benefit from this research. First, the IT consultancy company Accenture, which is involved in the research, could benefit from the insights the research provides. Accenture has many large IT service organizations as clients and many of them ask Accenture for advice regarding DevOps. They wonder what the DevOps way of working could mean for them. Therefore, the focus points for a DevOps implementation in such organizations can contribute to the advice Accenture provides to its clients. The knowledge on these focus points can benefit the implementation process of DevOps. When Accenture advises to a client to implement DevOps, they are often involved in the implementation step. It is for Accenture therefore key to make this implementation process also a success. Moreover, large IT service organizations themselves can benefit from the research results. The DevOps implementation model will support a IT manager of a large IT service organization to implement DevOps in their current Agile way of working. They can determine their implementation process of DevOps on basis of the relevant points to focus on, which support the success of the implementation process. This will increase the promised benefits of DevOps.

1.5. Research Methodology

Based on the research objectives, the main research question is formulated.

Research Question

How does an implementation model look like to support the implementation of DevOps in IT Service organizations?

To answer this research question, a Design Science Research (DSR) is performed. This type of research is chosen as the model can be interpreted as an IT artifact to support the IT manager with the DevOps implementation. The research methodology is based on the design cycle framework of Peffers et al. (2007). The design science research methodology of Peffers et al. (2007) “incorporates principles, practices, and procedures required to carry out such research and meets three objectives: it is consistent with prior literature, it provides a nominal process model for doing DS research and it provides a mental model for presenting and evaluating DS research in IS” (p. 46). This methodology is chosen as it is specially focused on building artifacts related to information technologies. It describes clear phases for the design cycle which are defined on basis of analyzing other DSR frameworks. Due to research constraints, such as time and resources, an adapted version of Peffers’ framework is used. The phases Problem Identification & Motivation, Objectives of Solution, Design & Development and Demonstration are included in the research methodology. Sub questions associated to the research question are formulated in order to structure the research within the phases. Figure 1 provides an overview of the research flow diagram which indicates the different phases and the chapter structure.

The first phase is “Problem identification and motivation” and is partly described here in Chapter 1 by explaining the situation, the problem and the objectives. To extend this phase with a more detailed problem identification and its context, the first sub question is formulated. This first sub question also covers the second phase “Objectives of Solution”.

1. *What is meant by a DevOps way of working and how is this related to currently used methodologies within IT service organizations?*

The problem identification is extended with a chronological description on the evolving way of working at IT service organizations. Different methodologies within such organizations over time will be

described, ending up with the need for a more DevOps way of working. There is no common definition on DevOps in literature and therefore it is necessary to develop a research specific definition. The answer to this sub question is retrieved by performing extensive literature research and conduct exploratory interviews with experts at Accenture. It will result in a research definition for DevOps and the key areas for challenges of its implementation. These challenges areas could be seen as the objectives of a solution, which relates to the second phase of Peffers' framework. The challenges areas are not specific requirements for the model, but imply the key areas which have to be involved in the implementation model, which make makes it possible to interpret them as the objectives of the solution.

The second sub question initiates the start of the third phase "Design & Development", which consists of three sections spread over sub questions 2, 3 and 4.

2. *What type of model for the implementation of DevOps should be used and how does a first design of the implementation model for a DevOps implementation look like?*

To formalize the implementation model, extensive literature research will be executed. Based on the definition of DevOps and its challenges, a theoretical approach will be chosen. Different theories regarding this approach will be used to design a theoretical model. By complementing this model with theoretical knowledge on DevOps aspects, a first conceptual design of the model is developed. This theoretical model for the implementation of DevOps can be seen as a first design of the IT artifact.

The outcome of the second sub question will be a theoretical model for DevOps in general, not yet specific for large organizations. The next step within this phase is develop this model further and make it more specific for the DevOps implementation at large IT service organizations. With this purpose, the developed is continued with formulating a third sub question.

3. *What is the value of the theoretical design for large IT service organizations based on empirical research and expert knowledge?*

By conducting case interviews at large IT service organizations, the value of the theoretical model for large IT service organizations is assessed. For each case at least one IT manager will be interviewed. This empirical research will indicate the relevance of the theoretical aspects for the organizations participating in the cases. The IT managers will be asked to answer multiple questions on their way of working and to indicate the importance of the aspects in the model. Possible additional findings during the cases will be evaluated on its correctness during an evaluation session with experts of Accenture. Based on the empirical findings of both empirical researches, final conclusions will be drawn upon the value of the theoretical design for large IT service organizations. This will contribute to the development of the model as this makes the model specific for such organizations.

The last sub question finalizes the "Design & Development" phase with developing the final implementation model and initiates also the begin of the Demonstration phase, which is the fourth phase of Peffers' framework.

4. *How does the new model look like for implementing DevOps in large IT service organizations and how can this contribute to the implementation of DevOps?*

Based on the conclusions of the third sub question, the design of the final implementation model will be developed. The design will be determined on basis of constructing building blocks that cover all relevant focus points and implementation aspects. These building blocks together can formalize a representation of the implementation model, which will be tried to show with a visualization of the final model. After the presentation of the final model, the development of the model is done and the "Design & Development" phase is finished. The use of the model will be demonstrated by illustrations that show the support that the model is giving to a IT manager of a large IT service organization. These illustrations are answering the second part of the sub question regarding the contribution of the model to the implementation of DevOps at a large service organization. This demonstration of the use of the model refers to the "Demonstration phase" of Peffers' framework.

Finally, the answers to all sub questions are summarized in order to answer the main research question. After that, the limitations of the research will be described and a reflection will be done upon multiple

areas. These two sections will provide input for future research. The research will finish with the recommendations to Accenture on the outcome.

In conclusion, the first four phases of Peffers' framework have been included in the research methodology. The other phases of the framework are out of the research scope due to time and resource constraints. The phases can easily be traced back in the research flow diagram (Figure 2), as well as the different sub questions which formalize one chapter each. In this way it is clear how the research and the document is structured and at which stage the research is while reading.

1.6. Chapter Conclusion

This chapter introduced the research problem, the research objectives and the research methodology of the research that is described in this report. The research problem is indicated as IT service organizations wondering how to implement DevOps. The research objective is therefore *“To develop a DevOps implementation model that can support IT managers of large IT service organizations, which have adopted an Agile way of working, with their process of implementing DevOps.”* The main research question which is based on this objective is *“How does an implementation model look like to support the implementation of DevOps in IT Service organizations?”*. To answer this question, a design science research is performed based on an adapted version of the framework of Peffers et al. (2007). The phases Problem Identification & Motivation, Objectives of Solution, Design & Development and Demonstration are included in the research methodology. Within these phases, associated sub questions are formulated. The structure of this report is determined based on these sub questions. Moreover, each chapter provides the answer to one sub question, although the phases are not bounded to the chapter structure as can be seen in Figure 1.

First, chapter 2 describes the chronological development of software development methodologies within IT service organizations which finishes with the emergence of DevOps. The research definition for DevOps is formulated and key challenges in relation with the current way of working are identified. Chapter 3 elaborates on the development of a first theoretical design for the model based on literature research. Next, chapter 4 starts with a detailed description of the used research methods to gather empirical data. The results during the case interviews and the evaluation session are discussed in detail and the chapter finalizes with the conclusions based on all empirical findings. Chapter 5 presents the formulated building blocks based on the conclusions of chapter 4. After the building blocks, the final model, an assessment tool and the illustrations are presented. Chapter 6 provides an overview of all sub conclusions as answers to the sub questions in order to answer the main research question. This chapter is also discussing the limitations, reflections, future research and recommendations to Accenture. Finally, the references are presented, followed by the appendices.

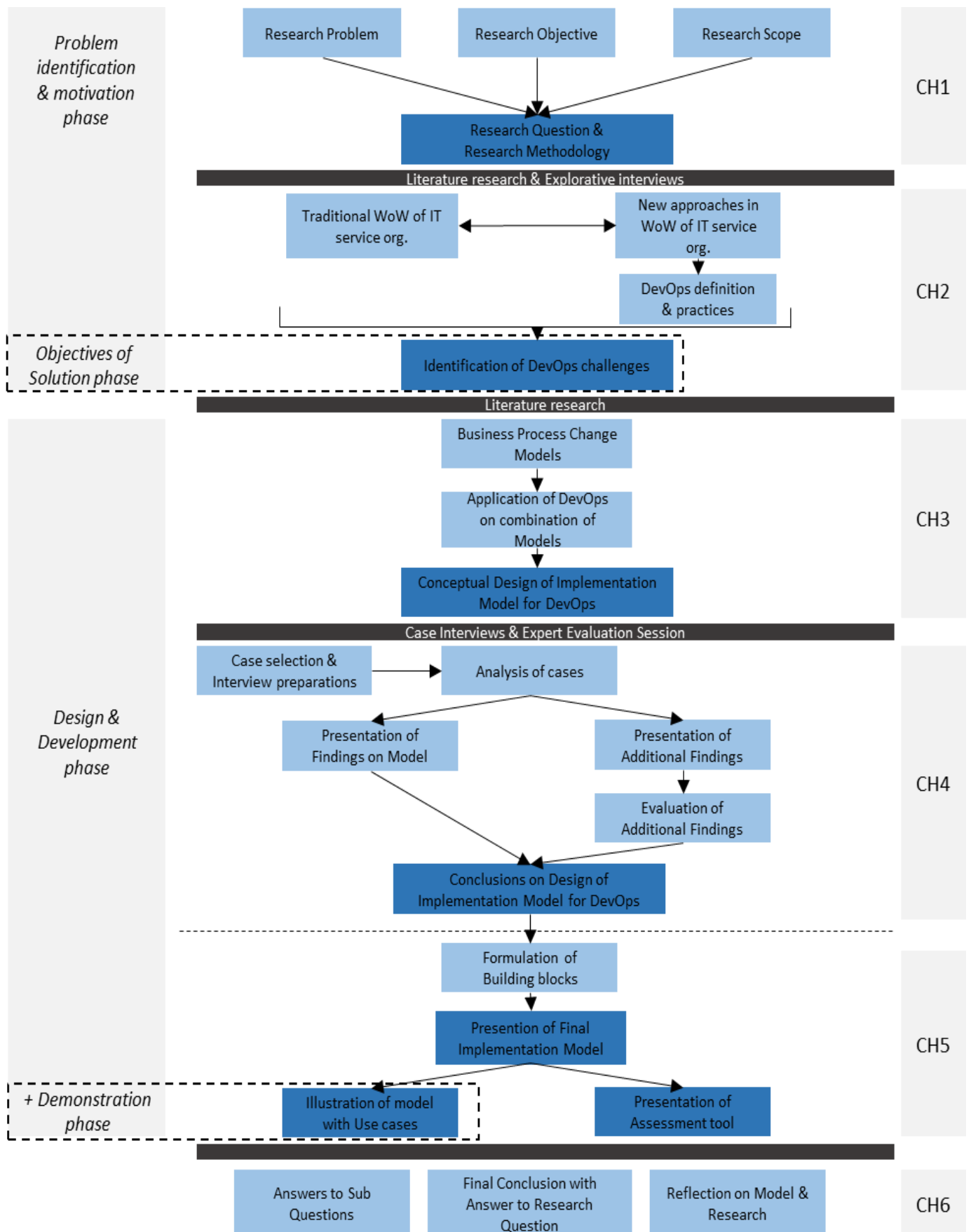


Figure 1: The Research Flow Diagram

Chapter 2

DevOps & Related Concepts

2. DevOps & Related Concepts

2.1. Chapter Introduction

This chapter provides the answer to the sub question “*What is meant by a DevOps way of working and how is this related to currently used methodologies within IT service organizations?*”. This sub question is formulated to create an in-depth understanding of DevOps and the challenges related to its implementation. This is an extension of the Problem Identification & Motivation phase, with a more in-depth analysis. By describing the older and current methodologies used within IT service organizations, it is possible to state the need for DevOps and the changes upon the current way of working. Based on this understanding, the research definition can be developed. This contributes to the next phase in which the theoretical basis for the model will be developed. Based on the objectives of the solution that will be defined in this chapter, the research approach on implementing DevOps can be decided in the following chapter.

The answer to the sub question is gained with doing extensive literature research and exploratory interviews which methods are explained in section 2.1.1. Section 2.2 presents a high level and chronological domain description of different methodologies that are used within such organizations, ending with the emergence of DevOps. Section 2.3 zooms in on DevOps and explains the perspective of the research on DevOps and its definition. This results in a deliberate description of DevOps for this research and an explanation of related concepts. The chapter finalizes with the key areas of challenges of DevOps that relate to the implementation of DevOps.

2.1.1. Research Method

The research methods that are used within this phase to answer the sub question are extensive literature research and exploratory expert interviews. Literature research was performed to gain knowledge for the whole research. The literature research gives a clear understanding of the different views on DevOps and the more theoretical approach. Together with this literature research, explorative interviews were conducted with experts at Accenture, as this gives some more insights on the practical knowledge on DevOps and the current situation within different organizations.

The main search engines for accessing literature was Scopus, ScienceDirect, Google Scholar and the search function of the reference program Mendeley. Books, articles, journals and conference summaries are used to gather information and more knowledge on the concepts. Keywords were used to find relevant literature such as: DevOps, development, operations, continuous delivery, continuous integration, continuous deployment, collaboration, information technology, systems, services, ITIL, ITSM, SDLC, Agile, waterfall method, incremental. The key words that are abbreviated here are written in full words during the search. When a useful article was found, the reference list was used for so-called ‘snowballing’ to find other useful references.

Also explorative interviews with experts at Accenture were held in order to gain more information regarding the current situation at organizations. Experts at Accenture have also a more practical interpretation of DevOps. They have gained much experience through projects at multiple large organizations about their current way of working and their possible ambitions on implementing DevOps. Based on the knowledge and experiences of the experts at Accenture, several experts were selected to be interviewed. The different experts with their role and experience is shown in Appendix A – Overview of Exploratory Expert Interview. The interviews were most of the time face-to-face or done through an online call. The length of the interviews was often between half an hour to one hour. The language of the interviews was Dutch or English, based on the native language of the interviewee. The questions that are asked during the short interviews are shown in Appendix B – Interview Guide Experts.

2.2. Domain Description

This section gives more insight in the delivery process at larger IT service organizations. First, in section 2.2.1 a general introduction on IT service organizations is given with defining such an organization and explain their high level delivery process. After that, a chronological description of different methodologies within this process is presented in section 2.2.2. The traditional methodologies will be

discussed first where after the emergence of other methodologies as improvement on the traditional ways of working are discussed. This chronological structure helps to understand the emerge of DevOps as these earlier methodologies can be seen as foundations on which DevOps is build.

2.2.1. IT Service Organizations & their Process

Current technology developments drive organizations to change their business strategy. Many organizations transform their business approach from delivering a service instead of a product. A service is defined by Hanna & Rance (2011) as “a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks” (p. 51). The emerging trend of offering a service is symbolized by companies such as Uber and AirBnB (Emidio et al., 2015). Both companies have won a large market share in short time by offering services of transport and places to sleep without owning any taxis or hotels themselves. This research focusses specifically on IT services, which can be offered as main business or used as supportive service next to a main business product or service. Initially, IT was adopted within business process to support the business process. Organizations perceived IT as just one of the many aspects within their business. However, the possibilities and the adoption of IT emerged over time. This lead to organizations that took IT apart from their main business and created separate IT departments (Interview 2). Currently, IT has become the backbone of the modern organization, resulting in offering many IT services. The strategic alignment model of Henderson & Venkatraman (1999) presents a clear overview of different roles of information technology in relation with the business (Figure 2). They state four different perspectives on the alignment between business and IT in which the influence of IT on the business strategy varies through these objectives. Although the influence of IT can vary, most organizations have still separated the business and IT management departments. These departments are often structured hierarchically with different departments with each its responsibilities.

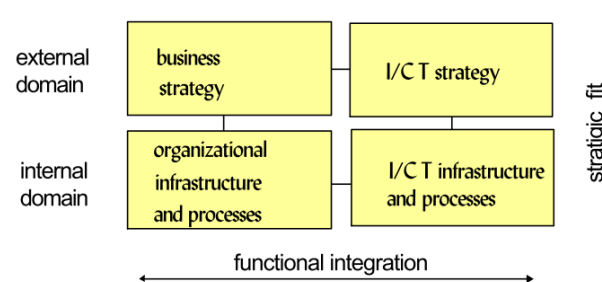


Figure 2: The Strategic Alignment Model of Henderson & Venkatraman (Maes, 1999)

2.2.1.1. IT Service Management

An IT service contains a combination of information, technology, people and processes (Bon, 2005). Focusing on IT services, ITSM methodologies are important. According to Galup et al. (2009), “ITSM is a subset of service science that focuses on IT operations such as service delivery and service support” (p. 47). There are many different frameworks within ITSM described in literature with each their own specific purpose and area. Some of the most applicable and widely used standards for ITSM are ISO/IEC 27002, COBIT, TOGAF and ITIL. Within organizations ITIL is the most commonly used framework for handling the overall IT service management. ITIL is paying attention to performing and organizing

more than one aspects of information management (Bon, 2004). ITIL contains some best practices found across the range of IT service providers and offers a framework for the delivery of quality of IT services (Bon, 2004).

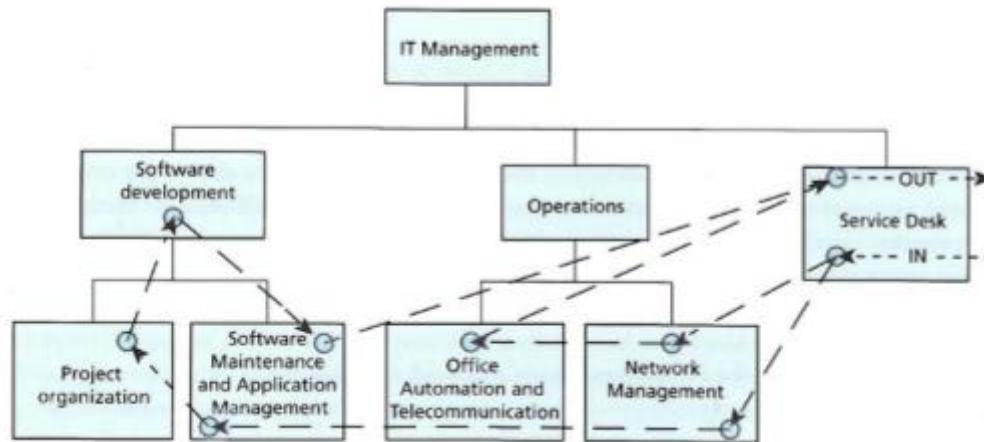


Figure 3: Example of processes between departments (Bon, 2004)

To structure an organization, functions and process are defined within ITIL. Functions are “subdivisions of an organization that is specialized in fulfilling a specified type of work, and is responsible for specific end results” (Bon, 2004, p. 55). An example of a function is a service desk. According to Bon (2005), “a process is structured set of activities designed to accomplish a defined objective” (p. 56). Figure 3 shows an example of a IT management department with its functions in which some processes are indicated by activities of different functional teams.

The strategic alignment model of Henderson & Venkatraman (1999) presents a clear overview of the role of information technology in relation with the business. To align IT service management with its business and their suppliers, different agreements are made between those parties. Between the business and IT service management, the requirements related to the customer are defined within SLAs. Hanna & Rance (2011) states that “an IT service supports the business processes directly and its service level targets are defined in a service level agreement (SLA)” (p. 32). To ensure the operational quality of an IT service, Operational Level Agreements (OLAs) are formulated between the suppliers of the IT service. As defined by Hanna & Rance (2011), “OLAs support the IT service provider’s delivery of IT services to customers and defines the goods or services to be provided and the responsibilities of both parties” (p. 39).

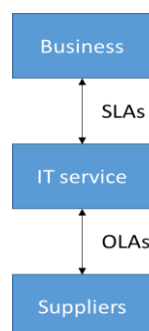


Figure 4: Position of SLAs and OLAs in ITSM

All services of an organization are described within a service catalogue. These services have their own service lifecycle which is described in ITIL V3 with the five phases: “service strategy, service design, service transition, service operation and continual service improvement” (Bon, 2004, p. 56). ITIL uses this service lifecycle to describe all relevant processes and functions of a service. The first phase is **service strategy** and focuses on the customer needs and what service will be introduced to the market according to these needs. The next three phases are centered around this formulated service and consists

of **service design**, **service transition** and **service operation**. These phases focus respectively on the requirements and the design of the new or existing service, on the building and deployment of this service and on the operational tasks regarding the service. The fifth phase is positioned around the other four phases as continuous improvement should be achieved within the whole phases. **Continual service improvement** indicates the importance of learning from past successes and failures and strive continuously for an increase of the effectiveness and efficiency of services and processes (Van Bon, 2004).

To stay competitive within the market, IT service organizations have to deliver their service more and more on a continuous pace. IT service organizations develop services by using a so-called delivery pipeline which can be compared to the ITIL service life cycle. The pipeline defines the activities necessary to create and deliver a service to an internal or external customer. These activities are often divided over the business department that formulates the business requirements and the different departments of IT, development and operations, which continue the process with subsequent activities regarding the development and implementation of a service (Figure 5).

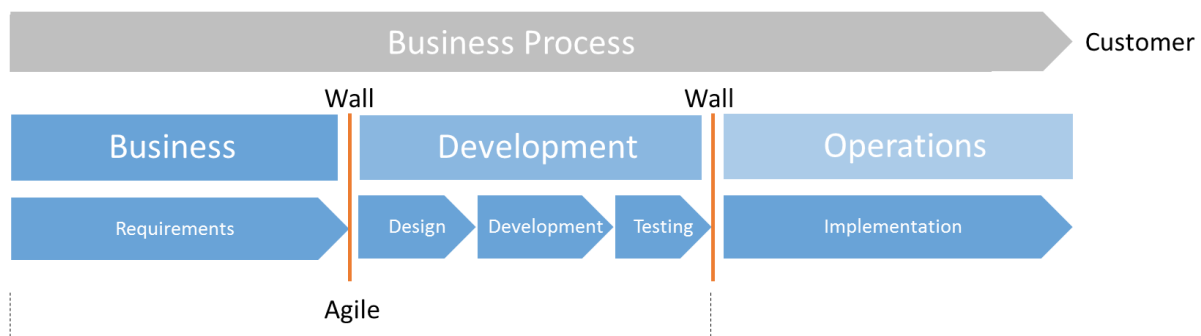


Figure 5: A high-level delivery pipeline of an organization

Several methodologies have emerged in order to make the process more efficient, decrease unnecessary waste and to structure the process. When focusing on the development, organizations have been moving from a waterfall approach to an Agile way of working with also focusing on a better alignment of the business requirements with the development. This evolution of these methodologies is described in the next section.

2.2.2. Evolution of Software Development Methodologies

The next sections will describe the evolution of software development methodologies that have emerged over time. Section 2.2.2.1 starts with describing the first software development methodologies that are known. After that, section 2.2.2.2 focusses on the Lean methodology on which many new methodologies are inspired. A more detailed description on the evolvement of the Agile methodologies is presented in section 2.2.2.3. Section 2.2.3.4 explains shortly the emerge of DevOps upon this Agile movement.

2.2.2.1. Earlier Software Development Methodologies

This section focusses on the earlier methodologies that are traditionally implemented at organizations. These methodologies will be discussed in a chronological order of emergence.

One of the earliest software development approaches is the waterfall method and has become a basis for most software standards (Boehm, 1988). The waterfall method is a sequential development model where each phase of the software development process must be completed before starting a next phase and there is no overlap between the phases (Wells, 2014). Figure 6 shows how this model follows the software development lifecycle, including the five phases business analysis, design, implementation, testing, and maintenance, see (Bassil, 2012). The fact that a phase should be completed before going to the next phase seems to make this model inefficient. The development process at organizations is taking often a few months and when the product is being released, it is possible that the customer needs are already changed (Interview 1). As a consequence of the inefficient model, several derivatives of the

waterfall method have raised, such as the V-model, the hybrid model and more incremental models such as spiral model of Boehm from 1988 (Mohammed et al., 2010).

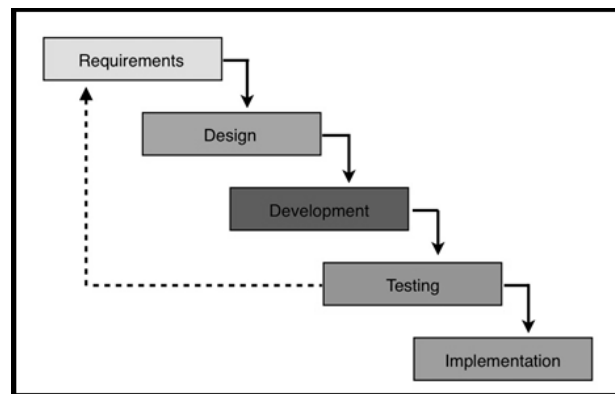


Figure 6: Software development lifecycle within the waterfall method

In order to make the development method more flexible a new method was emerging, called **Iterative Development**. This model divides a project in multiple subprojects that could be executed simultaneously, which help to give results earlier in the process (Mohammed et al., 2010). Although parts of the solution can be released incrementally, this method has some major disadvantages as for example more resources are needed and the project management becomes more complex. The **V-model** is similar to the waterfall model as it also consists of sequential phases that should be executed in a linear order. The only difference is that the V-model gives more attention to the testing phase (Mohammed et al., 2010). Verification and validation within the software development process is taking an important role according to the V-model, not only at the end of the development, but also before the development has started. This ensures that the design functionalities meet the requirements before the building starts. As the waterfall method consists of five phases, the **incremental model** that emerged as response has seven phases: “Planning, requirements, analysis, implementation, deployment, testing, and evaluation” (Bassil, 2012, p. 1). The **Spiral model** of Boehm (1988) can be seen as quite an innovative model for its time. It has emerged from multiple applications of the waterfall method to large government software projects (Boehm, 1988). The model focusses mainly on risk management and acknowledges already the importance of frequent and overlapping phases (Mohammed et al., 2010). The **Hybrid model** is also paying attention to risk management, but takes phase planning along as well. The main advantage of the hybrid model is that it can be applied to smaller projects.

2.2.2.2. Lean Thinking

Creating a IT service can simply be seen as a supply and demand process. Therefore, methodologies for improving such processes can also be applied at IT service delivery process. A methodology that has meant much for the development of a more efficient IT service process is lean thinking. **Lean thinking** can be applied to any system or process in order to indicate areas of improvement and applying such improvements (Hicks, 2007). According to Womack & Jones (1996): “it provides a way to specify value, line up value-creating actions in the best sequence, conduct these activities without interruption whenever someone requests them, and perform them more and more effectively” (p. 15). It is focusing on eliminating unnecessary waste in order to come closer to providing customers with exactly what they want (Womack & Jones, 1996). This is done with five categories of lean principles: value, value stream, flow, pull, and perfection (Staats et al., 2011). With applying the lean principles to the software delivery process, it can be seen as the basis for the movement of Agile development methods in 2001. Two principles that are interesting to mention here in particular are Kanban and the voice of the customer. **Kanban** is defined by Ikonen (2010) as “a flow control system for pull-driven production where upstream processing activities are triggered by downstream process demand signals” (p. 2). It is an effective tool to visualize the workflow, to see where work in progress (WIP) can be limited and to measure the lead time of the process (Kniberg, 2009). This can be seen as a similar approach to the Agile tool Scrum, where Scrum is focusing on sprints instead of workflows (Kniberg, 2009). Both approaches are paying much attention to the customer needs within the process. Also the concept the **Voice of the Customer (VoC)** as part of lean is assigning an important role for the customer in the process. It can be

described, as Hauser (1991) does, with “the tasks of identifying customer needs, structuring customer needs, and providing priorities for customer needs” (Abstract).

2.2.2.3. Agile Software Development

Partly based on the Lean Thinking principles, the concept of Agile software development has emerged and seems to be the most successful according to the degree of implementation at organizations. In 2001 the “Agile Manifesto for Software Development” was created (Beck, 2001). The manifesto consists of twelve principles for developing software with the four key values: individuals and interactions, working software, customer collaboration and responding to change (Beck, 2001). Under the umbrella of Agile software development are multiple practices and methods known, such as SCRUM, lean software development and eXtreme Programming (Dingsoyr et al., 2012). All methods under the Agile umbrella create a degree of agility in the process when it is applied. Agile software development practices have the characteristics of accommodating change in requirements at any stage of the development process (Dingsoyr et al., 2012). It is focusing on the phases business and development. The process is iterative with taking feedback into account so that customers’ needs are better satisfied. Applying Agile practices within the software development process can result in a decreasing lead time of software development (Interview 2).

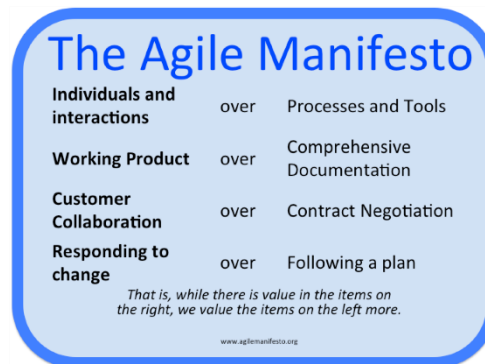


Figure 7: Basic principles of the Agile Manifesto (Beck, 2001)

The most widely used Agile practices are Scrum and eXtreme Programming (Yu & Petter, 2014). The **Scrum** approach is focusing on an iterative project within self-organizing development team with working in short sprints (Schwaber & Beedle, 2001). It assumes that the requirements are unpredictable and therefore the development should be able to adapt to changes in short cycles (Schwaber, 1995). At the end of each sprint an incremental prototype is delivered and compared to the requirements. With this feedback loops, the product can be adjusted as much as possible according to the customer needs and the velocity is increased. Self-organizing teams that are working with Scrum are often using an online or offline Kanban scheme to monitor the process (Interview 5). It can be assumed that Scrum is the most common practice within Agile. **eXtreme Programming (XP)** is on the other hand focusing much more on the process itself instead of the project management aspects. It consists of twelve practices and techniques, such as pair programming, that can help to improve the process (Barlow et al., 2011). The study of Fitzgerald & Hartnett (2005) showed that Scrum and XP can complement each other, with Scrum focusing on project planning and tracking and XP supporting the technical aspects.

Although, the implementation of Agile has proven to be beneficial, it also resulted in many organizations struggling with the transformation to an Agile way of working. Small and medium projects are often successful in applying Agile methods, however, larger projects have more difficulties with an Agile approach (Barlow et al., 2011). Larger projects cannot accurately estimate the needed resources and time without a detailed plan and Agile is also not focusing on formal communication and documentation among large teams and many stakeholders (Barlow et al., 2011). Therefore, there was a need for frameworks that scaled the Agile methods to enterprise level. Several different frameworks emerged for adopting Agile on a scaled manner (Melrose et al., 2015). One method is simply called **Large Scale Scrum (LeSS)** and is a framework that applies regular Scrum to large scale Agile development with

dividing major customer requirement areas (Larman & Vodde, 2013). **Scrum of Scrum** is seen as the method to handle inter-team coordination within LeSS (Paasivaara et al., 2012).

Other Agile methods to scale up are the **Scaled Agile Framework (SAFe)** and **Disciplined Agile Delivery (DAD)**. Besides the development of code, the purpose of both frameworks is to take also architecture, project funding, and governance of the processes and roles required by management into account (Melorose et al., 2015). At this level the very same lean and Agile principles that have worked well at the team level are applied. SAFe is scaling up the Agile method Scrum. It is focusing on the enterprise level with an organizational wide release planning session. This is similar to Scrum of Scrum, however, it coordinates the process more. A framework that is not extending only one Agile method, but is built around multiple Agile methods is DAD (Erich et al., 2016). According to Ambler & Lines (2012) the DAD decision framework is “a people-first, learning-oriented hybrid Agile approach to IT solution delivery, that has a risk-value delivery lifecycle, is goal-driven, and is scalable” (p. 4). The key aspects where DAD is focusing on are: solution focused, full delivery lifecycle, process goal-driven and enterprise aware (Ambler & Lines, 2012).

2.2.2.4. DevOps

Looking back to the whole delivery process, the fast velocity reached with applying Agile methods within software development does not result in a faster process. Teams of operations have to be able to cope with the fast lead time of development. In 2008 Tessem and Iden’s research gave insight into the limited cooperation of development and operations. The consequences of poor cooperation throughout the process are lower productivity in both phases, less quality of the software and service to users (Tessem & Iden, 2008). The interplay of development and operations is researched by Iden et al. (2011) and six most serious problems are indicated: “(1) IT operations not being involved in the requirements specification; (2) poor communication and information flow; (3) unsatisfactory test environments; (4) lack of knowledge transfer; (5) systems being put into production before they are complete; and (6) operational routines not being established prior to deployment” (p. 394). To avoid these problems, the importance of IT operations throughout the whole process must be acknowledged and the focus must be on enhancing cooperation and communication (Iden et al., 2011).

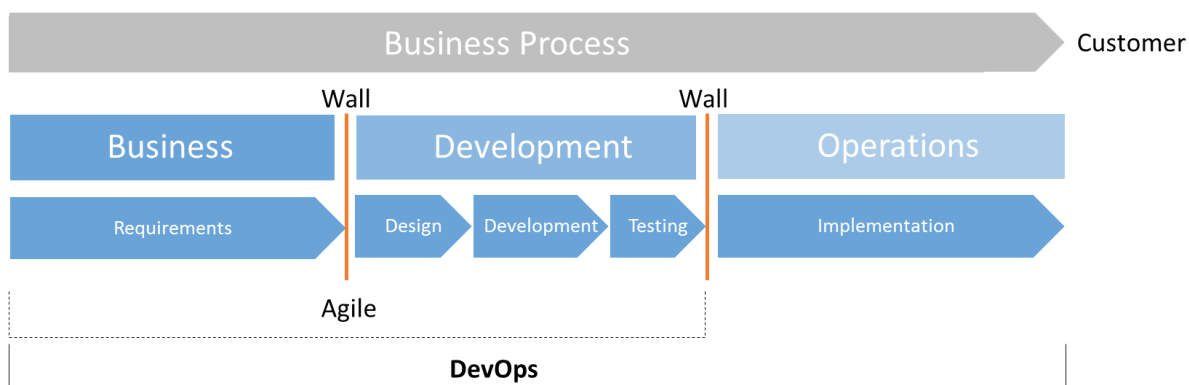


Figure 8: Indication of DevOps within delivery pipeline

In order to align development and operations with each other, the concept of DevOps is developed. DevOps can be seen as an extended version of the Agile software development approach as can be seen in Figure 8. Agile is focusing on integrating the business requirements with the software development within the IT process. It promises to result in cost reductions, high throughput of innovations, fast time to market and improvement of quality. Research has showed that working with an Agile methodology will make the software development phase more efficient and customer focused (Barlow et al., 2011). DevOps promises to take the Agile benefits to a higher level with taking also the operations phase into account and breaks through silos (Interview 2). Also according to Kim (2011) are the processes that emerge from DevOps an outcome of applying Lean principles to the IT value stream.

As DevOps breaks through silos and removes barriers, it often refers to the wall between the development and operations department. However, DevOps impacts more barriers, such as the important wall that is placed between the layers “application & infrastructure”, see Figure 9. Both are necessary for providing IT services, however people and systems of each functionality are used to a different way of working and rules. An interesting concept developed regarding this wall is called **Two-speed IT**. Two-speed IT proposes that Agile, innovative IT projects should be allowed to move forward quickly without being hampered by the necessary activities to maintain business-critical IT operations (Rouse, 2014). Therefore, the concept is describing two different speeds of IT projects within an organization. However, there is not much scientific research done regarding this topic which creates in still many discussions on this concept.

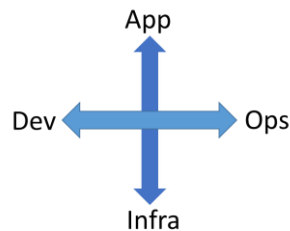


Figure 9: Overview of the walls of confusion

Some organizations have already undertaken some actions regarding DevOps. The choice for a specific organizational structure can already be seen as a beginning of the transition to DevOps. An organizational structure that organizations have adopted for DevOps is the **Spotify Model**. The Spotify model is already eliminating the friction between the two different departments. This structure is elaborately explained by Kniberg & Ivarsson (2012) and contains new terms such as squads, chapters and tribes (Figure 10). A squad is like a Scrum team self-organized and is end-to-end responsible for a part of creating business value. A tribe is collection of squads that have missions related to the same customer service. The method Scrum of Scrum can be used to coordinate interdependencies among these tribes, however, Spotify claims that the tribes are fairly independent and do not need a coordination meeting, only “on demand” (Kniberg & Ivarsson, 2012). The barrier between the development and operations is removed within the organization as the task of operations is to support the squads with releasing their code themselves, not to execute releases for them. Chapters contain each all people with the same skills and capabilities.

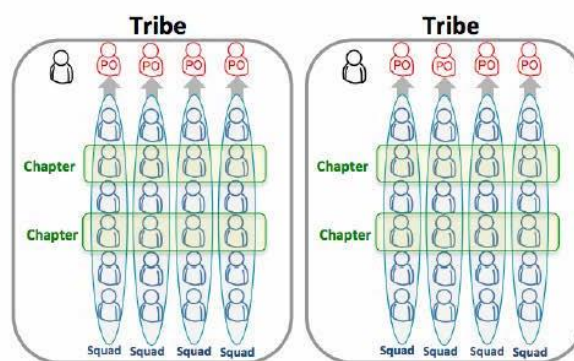


Figure 10: Spotify Model (Kniberg & Ivarsson, 2012)

With having DevOps raised as new “popular” trend, some criticism has emerged. People are questioning as within DevOps the operations team is integrated in the development team and teams are cross-functional, whether the operations department is still needed. This has resulted in again a new term **NoOps**. NoOps means to say that the development department is completely responsible for all relevant aspects of software production (Hüttermann, 2012). However, Hüttermann (2012) is stating that this scenario is not possible as certain tasks, both development and operations, still need to be done, independent of the team structure in which it is executed. When talking about fully DevOps teams,

where people are cross-functional skilled and can perform both types of tasks, it can be questioned whether the term operations is still applicable. Though, certain operations activities will always be necessary, independent of the fact people or automatic systems are executing them. For example, an organization will always need a call center to answer customers' questions, manually or automatically driven (Interview 2). This research is focussed on DevOps and approaches NoOps as a possible next step on which future research can be done.

2.3. DevOps

This section dives deeper into the concept of DevOps. A definition of DevOps is being established to take as starting point for the research (section 2.3.1). Section 2.3.2 provides the most important dimensions of DevOps. Section 2.3.3 discusses the most relevant practices of DevOps as DevOps practices are not commonly defined. Section 2.3.4 indicates the key challenges of DevOps. With knowing what a DevOps way of working includes, the research can be continued with researching how this way of working can be applied to larger IT service organizations.

2.3.1. DevOps Definition

DevOps is a combination of the words Development and Operations and it is a relatively new term within IT organizations. According to Hüttermann (2012), P. Debois mentioned the term in 2009 while organizing the DevOpsDays conference in Belgium. Erich et al. (2014) state however that the DevOps movement has been around since 2007. The term is gaining popularity within business, however, there is not one common definition for DevOps. When asking different people for the definition, the situation is similar to the famous poem of Saxe about six blind men and the elephant (Sato, 1927). Everyone has a different perspective on what the term exactly covers. In literature and online articles multiple opinions can be found which each argue it is a conceptual framework, organizational model, a philosophy, a mindset, platform, a group of concepts or just a tool. Therefore, it can be questioned who is right, however, it is clear that it is not just one tool or method that is implemented at once. Here are some of the definitions of DevOps found in literature:

Table 1: Overview of DevOps definitions found in literature

Author	Definition
Kim (2011) <i>Top 11 things you need to know about DevOps</i>	"The emerging professional movement that advocates a collaborative working relationship between Development and IT Operations, resulting in the fast flow of planned work (i.e., high deploy rates), while simultaneously increasing the reliability, stability, resilience and security of the production environment" (p. 4)
Bass et al. (2015) <i>DevOps: A Software Architect's Perspective</i>	<ul style="list-style-type: none"> ▪ "DevOps presents a fascinating interplay between design, process, tooling, and organizational structure" (Preface) ▪ "DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production while ensuring high quality" (p. 4)
Colavita (2016) <i>DevOps Movement</i>	"DevOps is a movement in the IT community that uses Agile/lean operations to add value by increasing collaboration between the development and the operations staff" (p. 203)
Bayser (2015) <i>ResearchOps: The case for DevOps in Scientific applications</i>	"DevOps (a portmanteau of "development" and "operations") is a software development method that extends the Agile philosophy to rapidly produce software products and services and to improve operations performance and quality assurance" (Abstract)
Erich et al. (2014) <i>Report: DevOps Literature Review</i>	"DevOps is a conceptual framework for reintegrating development and operations of Information Systems" (Abstract)
Erich et al. (2014) <i>Cooperation between software development and operations</i>	"The DevOps movement, a software development and operations professional's community, argues that the departmental division has led to cultural and communication problems" (p. 1)
Humble & Molesky (2010)	"DevOps is about aligning the incentives of everybody involved in delivering software, with a particular emphasis on developers, testers, and operations personnel" (p. 7)

<i>Why enterprises must adopt DevOps to enable continuous delivery</i>	
Walls (2013) <i>Building a DevOps Culture</i>	“A general consensus has started to form around DevOps being a cultural movement combined with a number of service development practices that enable rapid development” (p. 1)
Hüttermann (2012) <i>DevOps for Developers</i>	<ul style="list-style-type: none"> ▪ “DevOps is a mix of patterns intended to improve collaboration between development and operations. DevOps addresses shared goals and incentives as well as shared processes and tools. Because of the natural conflicts among different groups, shared goals and incentives may not always be achievable. However, they should at least be aligned with one another” (p. 9) ▪ “DevOps respects the fact that companies and projects have specific cultures and that people are more important than processes, which, in turn, are more important than tools. DevOps accepts the inevitability of conflicts between development and operations” (p. 9)

In line with Walls’ definition, DevOps is a cultural movement, including multiple practices, which aims for creating more flexibility and effectiveness within the business process. It breaks down silos and increases collaboration across different IT departments. The DevOps Culture exists of “open communication, incentive and responsibility alignment, respect and trust” (Walls, 2013, p. 15). Following a quote of P. Drucker who mentions that “culture will eat strategy for breakfast”, because a strategy or new business plan will never work when the appropriate culture is not provided among the organization’s people (Johnstone et al., 2011, p. 53). Also Hüttermann (2012) mentions the importance of people and the organization’s culture over the importance of processes and tools and Bayser (2015) acknowledges the importance of culture and communication as well. These statements are relevant for defining DevOps for this research. The research focuses on the implementation of DevOps within an organization with a current way of working. Based on the literature, implementing DevOps seems to affect the culture at the organization and its people at a high degree. Therefore, the cultural change is important to take into account and the research will take Walls’ description as starting point. However, the DevOps movement is not focusing only on people, it is taking all three aspects, people, processes and tools, into account. The definition that will be used throughout the whole research as basis is:

“DevOps is a cultural movement that breaks silos between the business, development and operations department, combined with a number of service development practices regarding people, process and technology, that enable rapid development and delivery of IT services”

2.3.2. DevOps Values

DevOps is extending the Agile methodology with taking its practices and align the ITIL processes used within operations with these practices. However, it is more than only aligning the practices together, it will also pervade through organizational structures and standards. Adopting a DevOps way of working seems to provide more benefits than just applying other approaches within either the development or operations phase. The most commonly shared benefits are considered as: cost reduction, high throughput of innovations, faster time to market and improvement of quality (Colavita, 2016).

Kim et al. (2013) describes DevOps principles by framing them with “The Three Ways” being: systems thinking, amplify feedback loops and culture of continual experimentation and learning. The article of Humble & Molesky (2010) is mentioning the importance of the concepts of the Culture, Automation, Measurement and Sharing (CAMS) framework. According to Erich et al. (2014) the way organizations implement DevOps depends on an extended version of the Culture, Automation, Measurement and Sharing (CAMS) framework with adding the concepts of services, quality assurance and structures and standards (Erich et al., 2014). Lwakatare et al. (2015) is also stating four different dimensions of DevOps, almost similar to the CAMS framework, consisting of Culture, Automation, Measurement and Monitoring. Based on (Pais, 2012)(Pais, 2012)Pais (2012), D. Edwards presented on the DevOpsDays of Italy in 2012 the four common pillars of organizations with a real DevOps vision, consisting of the three ways of Kim with added a fourth aspect: focus on flow. As stated before, DevOps is not just one

methodology that can be implemented. Neither is Agile one methodology, but consists of multiple practices and means that can lead to a more Agile process. The DevOps movement can be seen as an extension of Agile practices as it partially applies Agile practices to a wider part of the service delivery pipeline. It also involves elements of other methodologies, such as Lean Thinking and ITSM. For this research the first focus will be on the Three ways of Kim (2013) with the additional way of focusing on flow. These aspects are identifying the values of DevOps for this research.

Systems thinking means to focus on the performance of the complete system that brings value to the business, instead of just the performance of one department or silo. In this way the focus is on all business value streams that are enabled by IT, from the business or IT requirements to the end where the value is delivered to the customer as a service (Kim, 2013). Together with this, the interactions between the different phases within the process are taken into account. The so-called handovers between departments are creating walls of confusion among teams and the system approach is improving the interactions between different teams or people.

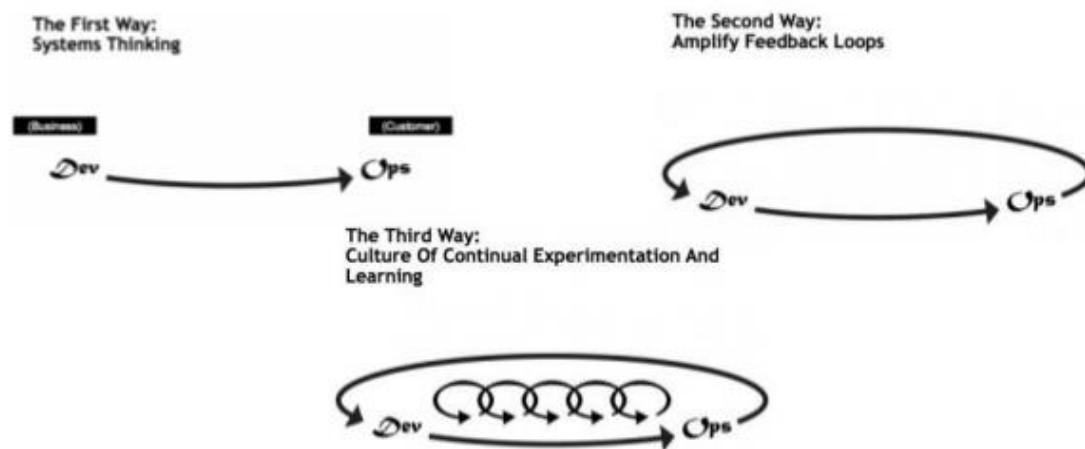


Figure 11: The Three Ways of DevOps (Kim, 2013)

Within the systems thinking idea, there is always sought for increasing the flow within the overall pipeline, which relates to the **focus on flow**. This is comparable to the supply chain theories of the bottleneck and the critical path. The lead time or flow of the process does not increase when only the performance of one aspect or phase is improved. It is important to approach the complete system from the beginning to the end. Also the importance of the interactions between different phases or departments are made clear with saying this. The famous quote of Aristotle is already acknowledging the importance of the interconnections with stating: “The whole is greater than the sum of its parts” (Goldstein, 1999). Improving only one specific phase or department is therefore not increasing the overall performance.

The second way of **amplify feedback loops** indicates the importance of feedback loops throughout the whole process. In this way necessary corrections can be continually made in order to response adequate to the customers, internal and external (Kim, 2013). This will also increase the understanding of the customer needs in order to respond quicker. These feedback loops relate to the Agile practices where short iterations and feedback is highly important.

The third way is focusing on the cultural aspect with being **culture of continual experimentation & learning**. Kim (2013) states the importance of creating a culture that embraces continual experimentation, taking risks and learning from failure on the one hand and understanding that repetition and practice is the prerequisite to mastery on the other hand. It is important to have a mindset of experimenting without regret within DevOps (Interview 5). By learning from the risks that are taken and master the skills that are needed to solve certain problems from the risks, daily work will be improved. Learning and sharing knowledge with each other is an important aspect within DevOps as this helps to understand the process and creates opportunities to improve it.

The four values of DevOps for this research are therefore: systems thinking, focus on flow, amplify feedback loops and culture of continual experimentation and learning.

2.3.3. DevOps Practices

Within these four dimensions, it is important to indicate which practices are covered by the DevOps movement. First of all, DevOps is seen as an extension of the Agile movement. Therefore, the Agile practices defined in the Agile Manifesto and also some of the characteristics of an Agile way of working will be the same for DevOps. As Scrum is the most common practice of Agile, the principles of Scrum are also taken for DevOps, such as focus on customer, cross-functional teams, feedback loops, short iterations and the importance of a working product. To provide a high-level overview of the differences of DevOps, an overview of multiple aspects is given in the table below. Some of the aspects of DevOps are based on the combined aspects of traditional and Agile methodologies, where others may be based on combining the knowledge retrieved from the literature research and interviews. An example for this is when considering DevOps from an Agile perspective, the core value of Agile “Individuals and interactions over processes and tools” (Beck, 2001) should be kept in high regard.

Table 2: Differences of waterfall, Agile and DevOps

Category	Traditional Adapted from Source: (Nerur et al., 2005, p.75)	Agile Adapted from Source: (Nerur et al., 2005, p. 75)	DevOps within this research
Fundamental Assumptions	<i>Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning</i>	<i>High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change</i>	High-quality, adaptive services can be delivered per multidisciplinary end-to-end teams using the principles of continuous improvement, integration, delivery and possible deployment
Role Assignment	<i>Individual – favors specialization</i>	<i>Self-organizing teams – encourages role interchangeability</i>	Multidisciplinary teams with end-to-end responsibility
Communication	<i>Formal</i>	<i>Informal</i>	Informal
Customer’s Role	<i>Important</i>	<i>Critical</i>	Critical
Project Cycle	<i>Guided by tasks or activities</i>	<i>Guided by product features</i>	Guided by delivering business value from end-to-end
Development Model	<i>Life cycle model (waterfall, Spiral, or some variation)</i>	<i>The evolutionary-delivery model</i>	Continuous Delivery Model
Desired Organizational Form/Structure	<i>Mechanistic (bureaucratic with high formalization)</i>	<i>Organic (flexible and participative encouraging cooperative social action)</i>	Learning (Interactive, flexible and collaborative people and teams)
Technology	<i>No restriction</i>	<i>Favors object-oriented technology</i>	Automation (Cloud, XaaS)

Zooming in on the DevOps practices extends the understanding of what is included for DevOps. Erich et al. (2014) uses the CAMS Framework as taxonomy to indicate the key concepts of DevOps. During their literature research they conclude to add three new concepts to the framework: Services, Quality Assurance (QA) and Structures & Standards. Systems thinking within DevOps requires a **culture** of collaboration. A DevOps culture is defined by Walls (2013) with the characteristics of “open communication, incentive and responsibility alignment, respect and trust” (p. 5). Continuous delivery provides an environment where changes and innovations can be adopted quickly and efficiently. To reach for continuous delivery **automation** of certain activities is key. Automation offers possibilities to make tasks more efficient (Erich, Amrit, & Daneva, 2014). Technical innovations drive the emerge of

new concepts such DevOps and are therefore heavily involved in its practices. Several new systems and applications are supporting the transition from manual tasks to activities done automatically. Organizations have currently already implemented many of these systems in their process. A new development that is gaining popularity within organizations is cloud computing. Following Hassan (2011), “**Cloud computing** adopts the concept of utility computing to give users on-demand access to computing resources in a very similar way to accessing traditional public utilities” (p. 16). Designing a business process with making use of the cloud provides organizations many new opportunities to face market volatility in an Agile and cost-efficient manner (Hassan, 2011). All kind of functionalities can be offered as a service which resulted in the term **Anything as a Service (XaaS)** of which examples are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) (Motahari-Nezhad et al., 2009). Therefore, automation is highly related to DevOps and with automation two important aspects should be taken into account, which are hiring skilled people that can work with the technology and as there are many opportunities possible, organizations have to trust employees with taking decisions and actions on this automation aspect (Erich et al., 2014). Both departments were measured in a different way which can lead to many frictions. Within the DevOps way of working this has to be changed and shared **measurements** have to be formulated. DevOps is about cross-functional teams, therefore **sharing** information, data and methods are very common. It is key that people document in their work and activities clearly to make it possible for other team members to work with it as well. Lwakatare et al. (2015) is replacing the fourth aspect of the CAMS framework with the aspect **monitoring**. Lwakatare et al. (2015) mentions the importance of “effective monitoring by emphasizing collaboration between developers and operations so that the systems are designed to expose relevant information” (p. 215). This is very similar to the sharing aspect, only Lwakatare et al. (2015) is also indicating the possibility of “using the provided information as feedback to developers and product management to use for product improvements and customization” (p. 216). The fifth aspect Erich et al. (2014) is giving is **service** in which area DevOps seems to be very beneficial with all XaaS concepts. **QA** is being able to do a more effective job as DevOps creates a chance for QA personnel to gain more data than they could in the past as departments are now working closer together (Erich et al., 2014). For a DevOps way of working, organizations have to consider change in their structures to facilitate the organization wide collaboration, therefore, the **structures and standards** should be taken into account during the transition (Erich et al., 2014).

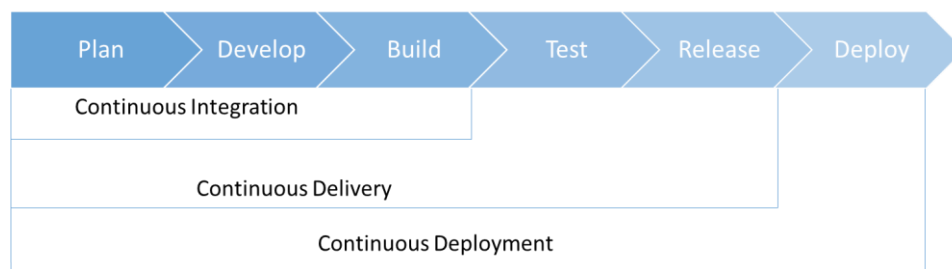


Figure 12: The software delivery pipeline with indication of different concepts

Next to that, it is relevant to zoom in on the meaning of the concepts continuous integration, continuous delivery and continuous deployment. These three concepts contribute to continuous improvement. As bringing value to customers is necessary for organizations, continuous improvement is always desired. According to Schaefer et al. (2013), “**continuous integration** means that there are no monolithic changes to the system configuration, but all updates ripple through the system in a series of small changes” (p. 346). Within DevOps it is desired to work according to continuous integration, moreover, it can be seen as a key requirement within DevOps. However, there is some confusion within literature about the fact whether DevOps is focusing on continuous delivery or continuous deployment. In order to take a point of view on this for this research, both concepts have to be defined. This can be done by looking at the pipeline that is being used with offering IT services. A lot of people are arguing that the term DevOps can actually be replaced by continuous delivery. Humble & Farley (2010) state that “**continuous delivery** is a way of working whereby quality products, normally software assets, can be built, tested and shipped in quick succession—thus delivering value much sooner than traditional

approaches and DevOps is a way of working whereby developers and IT system operators work closely, collaboratively, and in harmony towards a common goal with little or no organizational barriers or boundaries between them” (preface). Continuous delivery seems similar to DevOps, although DevOps is more focused on the complete organizational change than only on a continuous delivery pipeline (Humble & Farley, 2010). According to Bass et al (2015), DevOps is in many ways a response to the problem of slow releases. **Continuous Deployment** can be defined as deploying every change automatically to production whenever it is ready (Ten Hagen & Heunks, 2016). However, opinions differ on whether DevOps is only about continuous delivery and organizations should not aim for continuous deployment. Some state that continuous deployment should be the goal of organizations which are not constrained by regulatory (Caum, 2013). It can be questioned whether deployment should be desired and therefore it is interesting to take this into account during the research. The research will approach continuous delivery and continuous deployment pipeline as practices within a DevOps way of working as it covers the whole delivery pipeline.

2.3.4. Challenge Areas for DevOps Implementation

The literature and expert interviews show that DevOps is touching many different aspects within an organization. As Iden et al. (2011) stated, importance of IT operations throughout the whole process must be acknowledged. Therefore, there are many different areas where resistance can occur. Most organizations experience this resistance and face different challenges in the aspects **people, process and technology**. These three aspects are also known as the key elements for process improvement within organizations (Prodan et al., 2015). People, process and technology are often used by consultancy firms, also by Accenture, to indicate the dimensions in which changes or challenges for organizations occur that wish to implement a new way of working.

The most resistance is expected within the **people** aspect. An implementation of DevOps at an organization will have the most impact on the employees working in that organization. It requires a careful and comprehensive approach as people are among the highest importance of the organization. Following (Conboy et al., (2010), “the people issues uncovered include a broad range of problems from recruitment of Agile staff, to training, motivation and performance evaluation among others” (Abstract). Lwakatare et al. (2015) is stating the problem of poor communication between people, which can be indicated as limited shared knowledge between people. As people within their department are responsible for their own activities and deliverables, there is not a shared responsibility over the final deliverable. This creates the so-called silos where everyone is only focusing on their own part. Hüttermann (2012) is also stating three challenges among people that are similar to these problems: separate teams, no common language and fear. The last one, fear, can result in many areas as fear can occur as fear to lose power, influence or reputation. Next to this, people within an organization are common with certain processes, systems or processes and they often do not see the purpose of changing these habits (Hüttermann, 2012). People can also be resistant as they are afraid to lose their job, due to redefining roles or adopting automation (Interview 2). They will think their role will be redundant as unnecessary double roles exists when combining the two departments.

The process of Agile is focused on integrating the business requirements with the development. With DevOps the processes of operations are also integrated within this overall **process**. There are no separate phases anymore and the process has to be approached from the idea of systems thinking, referring to one of the three ways of Kim (2013). This integration requires some impacting changes within the process, such as storage and access of data and information, different interactions between systems, certain activities will be redundant or automatic instead of manual and KPIs or deliverables will be changed. The whole process has to be aligned with this new way of working and new KPIs seems to have to be applied. According to Erich et al. (2014), “organizations are afraid that these structural changes are incompatible with standards they use for their organizational processes” (p. 15). Lwakatare et al. (2015) is stating for example the problems of manual operational processes and performance of development and QA are not supported by data. Applying continuous delivery and continuous deployment means also a different process approach as frequency of releases and deploys becomes much higher.

When looking to the impact of adopting a DevOps way of working on the process within an organization that uses ITIL, it is important to acknowledge the fact that the processes defined with ITIL have to be aligned. Based on Interview 2 & Interview 3, it seems to be necessary to adjust ITIL to Agile instead of the other way around. It is stated that it is key is to find the most applicable way of adjusting these ITIL processes to the Agile way of working in order to let both development and operations cooperate. For example, for incident management, incidents are solved by different levels of operations. For incidents or requests that require only first level operations activities, it may not be necessary to integrate those within the whole process of development. Incidents on higher levels of operations seems to be solved more efficient when the operations team is working closer together with development team. Almost the same goes for problem management, where problems are indicated with problem levels.

The movement of DevOps is driven by technology developments. This indicates that the **technology** within an organization will also be affected by the DevOps way of working. New technology will facilitate many different opportunities within the organization and makes automation possible, for example virtualization and the cloud. Many managers are arguing that they cannot change their way of working as they have much legacy to cope with, such as current way of working and old technology and systems. Legacy systems within organizations creates a resistance within organizations towards the decision making step to adopt DevOps (Interview 2). It is a challenge to convince organizations of the long-term benefits of these new technologies to let them do the investment. However, a lot of people argue that this is just a matter of making an investment for the long term and organizations have to take this risk to stay competitive in the rapid changing environment (Interview 2). According to Erich et al. (2014) for organizations to stay competitive, it is important to build more heavily integrated products with sometimes exceeding the borders of an organization. Some people argue that you cannot work according to a DevOps way of working without automation in the IT systems. For instance, “instead of requiring the admins to manually search for errors in the network, a set of automated tests should verify the availability of all systems” (p. 346) as Schaefer et al. (2013) state. Also when the development phase is optimized to a certain degree, the delivery process can be even more efficient when tests are automated. Otherwise the developers have to wait longer than they need for development to get the test results back. Another possibility is an automated version control system. By ensuring the version are automatically updated, errors are expected to occur less and changes can be more easily applied (Schaefer et al., 2013). Automation is therefore very useful within DevOps, however, the way of implementation is a challenge, as also for adopting the Cloud within the business process. Managers within larger organizations are often stating that it is not possible for them, however, they are often focused on the short term and do not see the long term benefits (Interview 2). Applying the Cloud in an organization that is transforming to a DevOps way of working seems to be beneficial for its success. However, Melorose et al. (2015) is stating that the Cloud is not by definition necessary “as long as an organization has efficient processes for obtaining resources for deploying and testing application changes” (p. 62). During the research can be investigated in which degree the Cloud is involved in the large organizations.

2.4. Chapter Conclusion

The chapter answers the sub question: *What is meant by a DevOps way of working and how is this related to currently used methodologies within IT service organizations?*. First, an elaboration of the relevant way of working at IT service organizations is given, together with the related concepts and trends. Next, a research specific definition is defined and the practices of a DevOps way of working are explained. The research specific definition is:

“DevOps is a cultural movement that breaks silos between the business, development and operations department, combined with a number of service development practices regarding people, process and technology, that enable rapid development and delivery of services”.

The practices are mainly focused on the four dimensions: system thinking, amplify feedback loops, culture of continual experimentation & learning, focus on flow. Also an overview of DevOps characteristics is given in comparison with other software development methodologies.

This chapter contributes to the research as the field in which DevOps operates is defined. This gives a clear direction to the whole research which helps to focus on the scope during the research. Also, the areas in which the key challenges appear are similar to the elements of the process improvement model of Prodan et al. (2015): people, process and technology. Therefore, these three areas have to be involved in the implementation model that is being developed and can be seen as objectives of the solution. With having these objectives the Design & Development phase can be started.

Chapter 3

Design of the Theoretical
Model

3. Design of the Theoretical Model

3.1. Chapter Introduction

This chapter provides the answer to the third sub question “*What type of change model for the implementation of DevOps should be used and how does a first design of the implementation model for a DevOps implementation look like?*”. This sub question indicates the start of the Design & Development phase of the research methodology. To be able to start designing an implementation model for DevOps, a starting point from theoretical perspective is necessary. This helps to retrieve a rigid basis for changing or improving the current way of working at an organization. By answering this question, the theoretical approach of the research is chosen based on the objectives of the second chapter. This general approach is adjusted to fit the characteristics of DevOps in order to suit the objective of developing a DevOps implementation model for large IT service organizations.

First, the theoretical research approach is explained and multiple business process change models are discussed based on literature research (Section 3.2). The theoretical models serve as input for the structure of a conceptual model. Secondly, in section 3.3, literature on DevOps provide the DevOps specific requirements for the conceptual model. At the end of the chapter a conceptual model is developed which can be seen as a first design of the implementation model for DevOps.

3.1.1. Research Method

Extensive literature research is used within this phase to answer the sub question. As a theoretical basis is required, literature research is suitable to discover the most relevant scientific theories on general change or improvement models. Together with the literature gathered during the first literature research on DevOps, the selected theories are adjusted towards DevOps. Key words related to Business Process Change and DevOps were used, such as: development, operations, continuous delivery, continuous integration, continuous deployment, collaboration, information technology, systems, services, business process change, reengineering, TQM, Lean, Six Sigma, information technology. The main search engines for accessing literature was Scopus, ScienceDirect, Google Scholar and the search function of the reference program Mendeley. Books, articles, journals and conference summaries are used to gather information and more knowledge on the concepts. When a useful article was found, the reference list was used for so-called ‘snowballing’ to find other useful references.

3.2. Process Change Model

This section discusses the multiple business process change models that are found in literature. These models contain all relevant elements which can be used to design a conceptual model for the DevOps implementation at large IT service Organizations. Therefore, the theoretical basis for the model is designed with these models. First, the choice of business process change field is explained (3.2.1). Section 3.2.2 describes the revolutionary approach of business process reengineering. Section 3.2.3 discusses the more evolutionary approaches Total Quality Management and Lean Six Sigma. Section 3.2.4 explains which elements of the models are chosen and presents the structure of the conceptual model.

3.2.1. Business Process Change

Following the conclusion of chapter 2, the key areas of challenges for DevOps are people, process and technology which are similar to the process improvement model of Prodan et al. (2015). Based on this and following Swartout (2014), adopting a DevOps way of working in an organization can be approached as a business process change. A business process can be defined as “a lateral or horizontal organizational form, that encapsulates the interdependence of tasks, roles, people, departments and functions required to provide a customer with a product or service” (Earl, 1994, p.13). This research approaches the implementation of DevOps as a business process change as well for the design of the implementation model. With implementing DevOps, the focus on value is adopted in the whole business process by breaking through silos and focusing on system thinking throughout the whole process. It is not limited to only the development and operational department. It is also not limited to the software

delivery process, the way of working can also revitalize other business processes (Swartout, 2014). Therefore, business process change models can be used to approach the implementation.

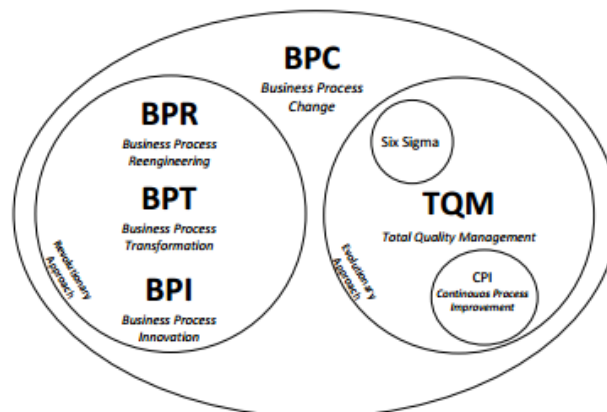


Figure 13: Overview of all BPC concepts (Kristekova & Jurisch, 2012)

Kettinger & Grover (1995) define Business Process Change (BPC) as “a strategic-driven organizational initiative to improve and (re)design business processes to achieve competitive advantages in performance through changes in the relationships between management, information, technology, organizational structure and people” (p. 12). BPC comprises both radical or revolutionary and incremental or evolutionary management approaches as can be seen in Figure 13 (Kristekova & Jurisch, 2012). These two-sided division of approaches relates to perspectives of top-down and bottom-up. Organizational change can be approached from two perspectives, top-down or bottom-up. The concepts of these approaches are quite similar to strategies of push and pull. A top-down approach starts with a policy decision by governmental officials, often the highest managers within the organization, and pushes the policy through the whole organization (Sabatier, 1986). Bottom-up means to start with an analysis of employees or teams who interact at the operational level on a particular problem or issue and let the change emerge from them up to the higher organization. Based on these definitions, DevOps seems to be a movement originated from a problem within hierarchical lower teams. However, the decision of implementing DevOps seems to be often strategy driven. Using both approaches in the research makes it possible to analyze which approach is most suitable for DevOps. As the implementation of DevOps is approached in this research as a change within the business process, business process change models are used to research on which aspects the implementation should be focused and how this can be done within large IT service organizations.

3.2.2. BPR

The **Business Process Reengineering (BPR)** methodology is one of the revolutionary approaches together with business process transformation and business process innovation. It is a relatively old approach, however, still strong as a basic foundation for reengineering processes. According to Guha et al. (1993), “business reengineering seeks to redesign work processes to enhance productivity and competitiveness” (p. 14). Hammer & Champy (1993) have a more extensive definition which is “fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical contemporary performance measures, such as cost, quality, and speed” (Hammer & Champy, 1993, p. 32). It is often stated that BPR is a top-down approach as it assumes that a change has to be strategic driven to be successful (Kristekova & Jurisch, 2012). BPR frameworks comprise multiple factors of a process that are impacted with a change.

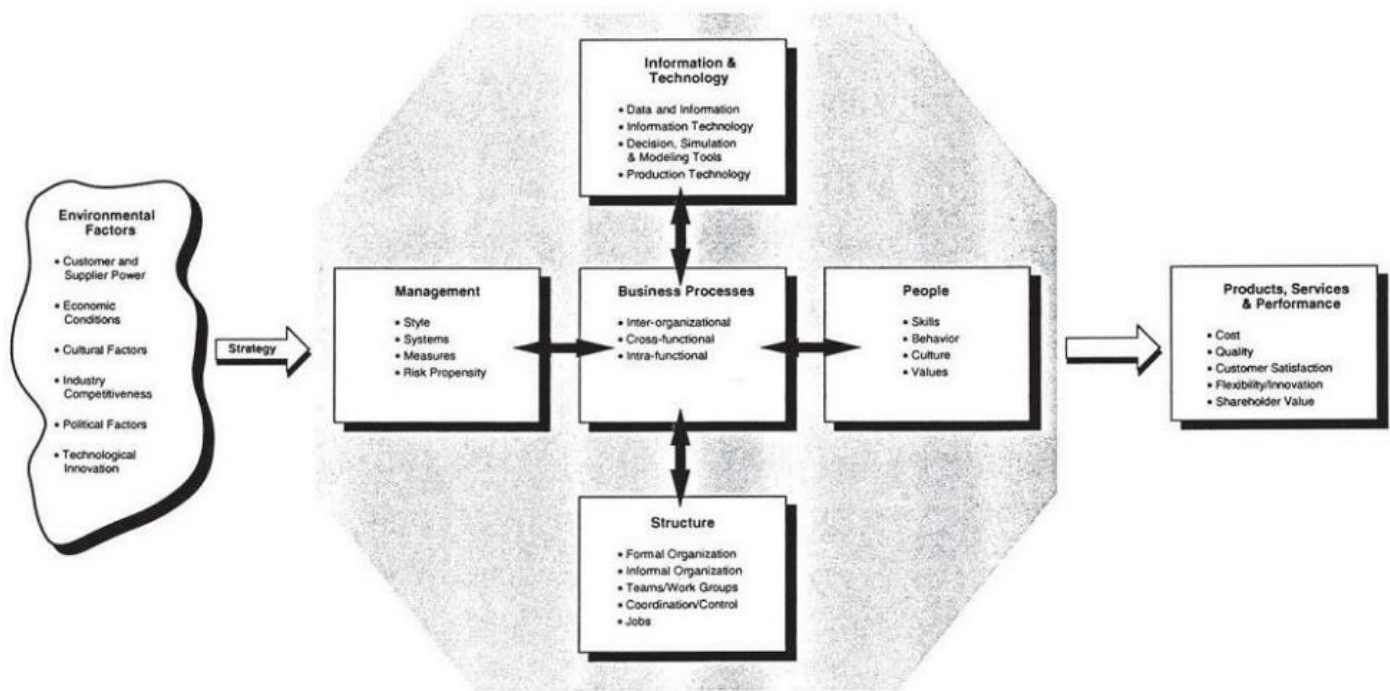


Figure 14: BPR framework (Kettinger & Grover, 1995)

One of the most dominant frameworks is the business process change model of Kettinger & Grover (1995) that provides a clear overview of transformational subsystems of BPR. They make a division of the subsystems: business process, management, information & technology, people, organizational structures (Figure 14). The restructuring framework of Mayer & Benjamin (1995) is also providing multiple dimensions within organizations that are impacted by BPR. Changes cannot result in performance improvements if only one of these dimensions is touched, which means that a change requires an integration of the dimensions. They state six dimensions: control, culture, configuration, people, technology and information (Mayer & Benjamin, 1995).

Another model, the integrative model for IT-enabled business process change of Jurisch et al. (2012) states that the organizational culture & structure is outside of the change initiative, but indicates the importance of IT and human resources, project and change management as well. BPR comprises different techniques and tools to lead a BPR project within an organization. Kettinger & Grover (1997) have researched many of such techniques and tools and give a clear overview of the comparison of them. However, because of a high failure rate of BPR projects, scientists are stating the BPR approach is too radical (Weerakkody, Janssen, & Dwivedi, 2011). Business process change management has to be done along the process, but it can be questioned if this is enough to avoid failure.

3.2.3. TQM / LSS

Another model that is often placed in contrast with BPR is **Total Quality Management (TQM)**. TQM has both philosophical elements as well as management elements for an organization to improve quality (Love et al., 2000). Improving the quality within TQM is focused on improving business processes incrementally and it acknowledges the importance of the customer. TQM is taking a bottom-up approach with employing continuous improvement activities and having an understanding of the process (Weerakkody et al., 2011). TQM elements are often seen as: customer focus, total employee involvement, process-centered, integrated system, strategic and systematic approach, continual improvement, fact-based decision making and communication (Fotopoulos & Psomas, 2009).

One of the practices related to TQM is Six Sigma. Six Sigma is a structured and systematic approach to process improvement, using statistical techniques to make fact-based decisions (Pepper & Spedding, 2010). Six Sigma is providing an answer to the weakness of TQM of being only a philosophy by adding business metrics to the quality elements. As such concepts have been implemented in isolation, sub cultures of methodologies have emerged within organizations. To integrate these methodologies more in organizations, the concept of Lean Six Sigma (LSS) has emerged (Figure 15). According to Pepper & Spedding (2010), “aligning the cultural aspects of Lean with the data driven investigations of Six Sigma holds huge potential in a bid for a genuine and sustainable approach to organizational change and process improvement” (p. 151). Applying LSS initiates a change of cultural and operational change which will eventually change the total supply chain. It results in a more integrated, coherent and holistic approach to continuous improvement (Pepper & Spedding, 2010).

Concept	Explanation
Lean	A culture using tools aimed at minimizing waste and creating more value while doing less work
Six Sigma	A data-driven approach to error reduction by improving processes and reducing process variability

Key Principles of Lean Six Sigma	
•	Focus on the customer (i.e., the patient and/or physician)
•	Identify and understand how work in the lab gets done (the value stream)
•	Manage, improve and make workflow more efficient
•	Remove non-value-added steps and waste
•	Manage the lab using data and reduce workflow variation
•	Empower (involve and equip) the people in the process
•	Systematically undertake improvements in all activities

Figure 15: Lean Six Sigma (Cucoranu, Parwani, & Pantanowitz, 2014)

3.2.4. Theoretical Design of Model

In summary, the business process change methodology is used to develop a conceptual model for the implementation of DevOps. The concepts of BPR, TQM and LSS are discussed in the previous sections. The BPR indicates that a change should be strategic driven and provides clear categories that are impacted by a change. LSS provides a holistic approach to continuous improvement. DevOps is built upon these methodologies and shares several practices. The conceptual model is being built with using a combination of the frameworks.

The structure of the BPR-model of Kettinger & Grover is taken as the basis of the model. It provides a clear and general overview on how a business process change is driven by **environmental factors**, which subsystems and its interrelations are impacted and which **performance measurements** are important. To identify the subsystems that are relevant for DevOps, the DevOps dimensions and challenges are compared with the defined subsystems of multiple BPR-models. The challenges of a DevOps implementation are stated within the three aspects **people**, **process** and **technology**, which matches the holistic model for Process Improvement (Prodan et al., 2015). Bon (2004) also states that people, process and technology is a widely accepted paradigm for the focus areas in organizational improvement. The reasons for occurring conflicts between the current and new way of working are often caused by differences in the organizational culture, skills and mindset of people and the processes (Bang et al., 2013). The category process includes all process elements used within the overall business process. Technology refers to all tools and techniques that are used within the business process to communicate and to make the process and its flow more efficient (Prodan et al., 2015). **Information** is added as an extra category, separate from technology. Another model, the Pearson model, acknowledges the importance of information, according to Prodan et al. (2015) and also Bon (2004) states that information is highly related to these three aspects. The information shared and available across the

whole business process within IT service management is crucial. The category **organizational culture and structures** is positioned over all main categories, similar to the integrative model of Jurisch et al. (2012), as this relates to the environment in which process, people and technology interact (Prodan et al., 2015). Prodan et al. (2015) even argue the existence of three new dimensions to the people, process and technology mode which are customer focus, innovation and management functions. These dimensions are also related to DevOps, but those aspects are categorized under the other categories and not as separate dimensions. The aspect management can be traced back within the aspects organizational culture and structures, process and people. Although management of the overall process is important, it is relatively lower as the teams are aimed to be more self-organized and responsible.

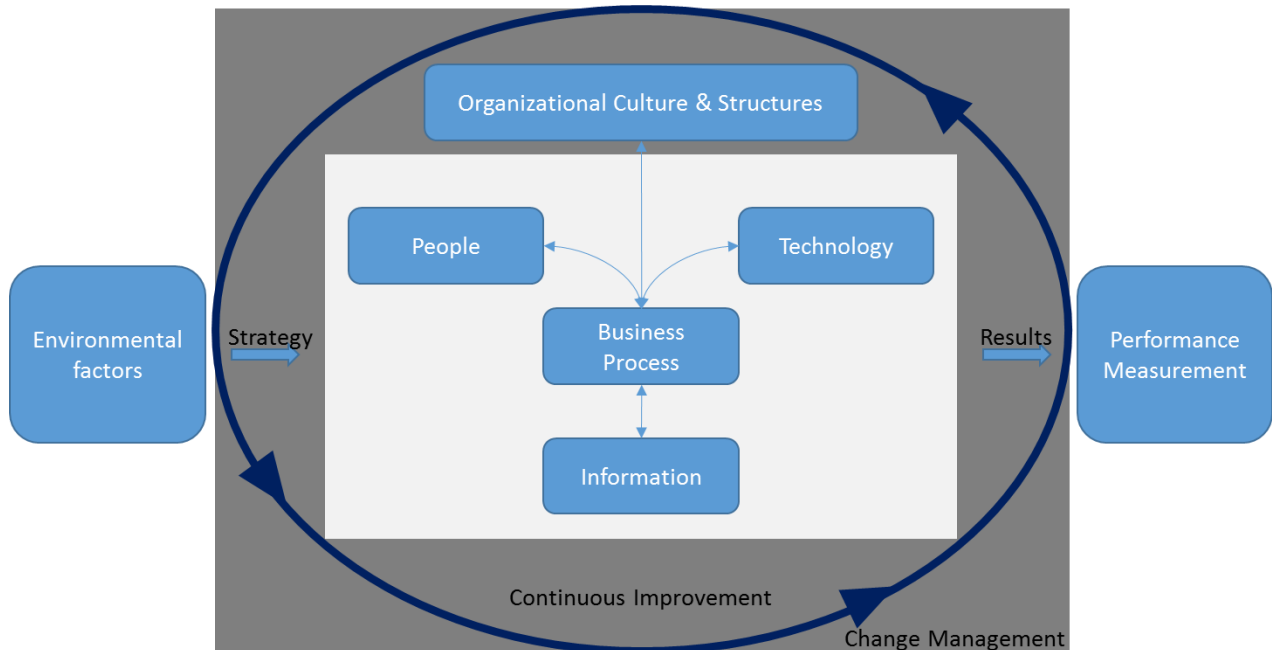


Figure 16: Theoretical design of model

The influence of the other evolutionary methodologies can be seen by the **continuous improvement** circle and within the aspects under the categories, described in more detail in section 3.4. Continuous improvement is important in the overall business process as well as in each category on its own. In fact, without a continuous improvement process, reengineering cannot be successful (Attaran, 2004). The bottom-up approach is also indicated with the circle around the categories. Change can be initiated or led by an aspect within one of the categories. The **change management** that will guide the actual change is taken into account as a separate category. Organizational change management is defined by Moran & Brightman (2000) as “the process of continually renewing an organization's direction, structure, and capabilities to serve the ever-changing needs of external and internal customers” (p. 111). Management of these changes should be taken into account as the implementation of a new way of working will never have a chance to be successful without managing the changes carefully (Grover et al., 1995). Everything within the continuous improvement circle is considered as a project scope and matches the research's scope.

All aspects are viewed from the perspective of the IT manager. The categories environmental factors and high level performance measurements are an exception, as these are often not determined by the IT manager, but from a higher level manager or CIO. The environmental factors influence the strategy that determines that changes of the categories within the circle and the achieved changes are communicated back to the high level performance measurements. The overall organizational culture & structure is not managed by only the IT manager himself, only on a lower level he can influence this. These categories are still important within this research to gain knowledge on the purpose and measurements of DevOps in an organization. However, within the research the main focus will be on the other categories. The structure of the conceptual model is extended in more detail in the next section.

3.3. Theoretical Model for DevOps

This section develops the conceptual model in more detail. The structure from section 3.2 is extended with more characteristics of DevOps on each of the categories defined in the model. First, section 3.3.1 zooms shortly in on the environmental factors and section 3.3.2 on the performance measurements. The other sections explain the main categories. Section 3.3.3 describes the aspects related to the process. Section 3.3.4 describes all aspects regarding people and section 3.3.5 describes the aspects regarding to information. Section 3.3.6 discusses the aspects of organizational culture and structures. Section 3.3.7 zooms in on technology and section 3.3.8 provides aspects for the change management category.

3.3.1. Environmental Factors

Although it is not the main focus point of the research, it is good to know what the IT managers interpret as the reason for changing the current way of working towards a DevOps way of working. Especially to research whether the driver for DevOps is coming from top-down or bottom-up. Therefore, the IT managers will have to indicate which of the following factor(s) they see as reasons to implement DevOps way of working: **External Customer and Supplier Power, Economic Conditions, Industry Competitiveness, Political Factors, Technological Innovations**. These factors are given by the model of Kettinger & Gover (1995) and are used to retrieve a high level interpretation of the external drivers for adopting DevOps. The external customer can demand a faster service. Customers could give this as feedback, but it can also be noticed by an organization when the sales are not satisfying. The supplier of an organization has also much power. When a supplier's way of working is faster than the organization, this could be a driver to increase the lead time of the delivery pipeline. However, the supplier can also negatively affect the process of an organization. When an organization desires to deliver at higher speed, it can be limited to the supplier's way of working as it is depending of certain tasks or material of a supplier. Competitors of an organization can also drive an organization to change their way of working. When a whole industry increases its time-to-market, it is important for an organization to stay up to date to its competitors to sustain their market share. As DevOps involves many new technology practices, technology innovations drive organizations probably highly towards the implementation of DevOps. The economic conditions and political factors does not immediately seem to drive the implementation of DevOps at a high degree. Prices, costs, laws and rules do have effect on the way of working, though not directly related to DevOps specific. However, by taking those factors into account in the model, it can be discovered whether these factors are indeed not highly relevant.

3.3.2. Performance Measurements

For the same reason as the environmental factors, the performance measurements are questioned at the level of IT managers. With this information it can be analyzed which performance metrics are used at business and team level and whether this is already in line with an Agile or even DevOps way of working. For this purpose, the factors of the Kettinger & Gover model is taken, except of shareholder value, as those are also the most relevant for IT managers. It is assumed that shareholder value is hard for IT managers to analyze.

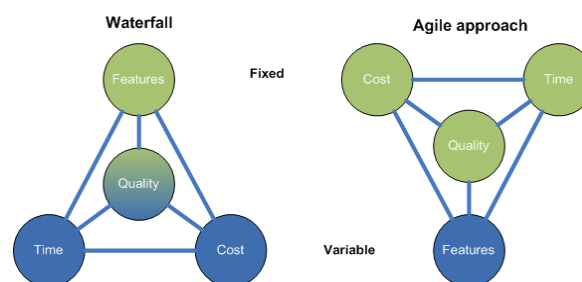


Figure 17: Overview of fixed and variable measurements within Waterfall and Agile

For a new way of working the performance measurements systems have to be adjusted for a successful adoption of Agile (Nerur et al., 2005) and also Erich et al. (2014) states that traditional employee

performance measurements lead to friction. Within the Waterfall method the features are fixed as the requirements are set by the business and these requirements determine in the beginning of the process what has to be done. The time and cost are variable and the quality also partly. The quality is for a certain degree set in the requirements, however, as the process is structured as a waterfall, the quality at the end of the process cannot be guaranteed. Within the Agile approach, this is the other way around. The cost, time and quality are fixed and clearly formulated in the beginning. Only the features that are being developed are variable within those fixed aspects. Therefore, the factors time and capacity / velocity of the teams are added as these factors are related to waterfall and Agile business processes. The capacity / velocity of the team is the speed on which a team develops features. The Flexibility / Innovation indicates the ability of the team or process to react to changes or new developments. It is useful to discover whether these factors are used within their current way of working, from both business and team perspective. In conclusion, the factors **Quality, Cost, Customer Satisfaction, Flexibility / Innovation, Time and Capacity / velocity of the teams** are adopted in the model.

3.3.3. Process

The business process of software development consists of many sub processes. DevOps principles relate to the overall business process as well as those smaller processes that are defined at lower levels. First, the concept of **system thinking** as main dimension of DevOps is important over the whole process of IT services (Kim, 2013). Within this approach, there are many methods that are known from the Agile methodology used across the whole process. As the research is focused on the IT manager and its scope, this relates to the fact that the delivered product is viewed holistic by everyone involved. In this way the objectives and interests across the whole process are aligned and no silos are present. This approach results in the emerge of **end-to-end responsibility within self-organized teams**, where self-organizing teams are already described within Agile Manifesto (Beck, 2001). In DevOps teams are required to work on their own product or feature and have the responsibility on the whole process of their product from defining the requirements to maintaining that product or feature (Hüttermann, 2012). System thinking allows an organization to **focus on their customer**. The requirements can be prioritized for each feature according to the customer. This is also reached by using Scrum for example, but DevOps is extending this by involving operations in the system approach as well. To enable this, short **feedback loops** with customers, external but also internal, have to be realized. Feedback loops have to be used within the teams as well, by working in similar manner as Scrum. Working in **short iterations** like sprints create the ability to change quickly and respond to the customer's demands.

Next to these aspects, which are quite similar to Agile methodologies, there are some process aspects that are important to a DevOps way of working. First of all, it is argued by Iden & Tessem (2011) that "IT operations is not being involved in the requirement specification and operational routine not being established prior to deployment" (Abstract). In order to solve these problems, the operations department have to be more involved in the entire process. Therefore, **operational processes (ITIL) have to be aligned with the Agile process of development**. When these processes are aligned with each other, it is easier to create cooperation among the different tasks in the phases. According to (Interview 2), the ITIL processes have to be aligned with the Agile process and not the other way around. However, this is from the view point of an operations manager. Eventually, when the processes are aligned and one team is created, there have to be an appropriate **balance between innovation and problem management**. Innovation includes here also product development. This can for example be done with a prioritization of activities that have to be done and treat the two categories of activities within a fair amount of capacity or effort. Besides this, **non-value adding activities have to be removed** according to the Lean methodology. With having this mind-set, more efficiency is established and unnecessary or redundant work is eliminated from the process. **Continuous Integration (CI)** can be seen as a critical mean to enable DevOps. Continuous Integration means that there are no major updates to the system configuration, but updates are only possible in series of small changes in the system (Schaefer et al., 2013). It ensures that the code that the team creates works by providing rapid feedback on any problem that may be introduced with the committed changes (Humble & Farley, 2010). However, CI is only focused on the development phase. **Continuous Delivery**, with having on-demand releases can be counted as a next step and is a highly important factor of DevOps. The term is already described within the Agile Manifesto: *"Our highest priority is to satisfy the customer through early and continuous*

delivery of valuable software” (Beck, 2001, p. 2). Moreover, it is even more realized with a DevOps way of working as with DevOps the whole delivery pipeline is focused on delivering value continuously. Continuous Deployment is related with this term and will be explained further under the technology category.

3.3.4. People

This category seems to have a high importance. Although a process view is taken in the research, the focus on people and their capabilities is crucial for a DevOps transformation. The interviews with the experts showed that a change in the way of working can only be successful when people are cooperating and willing to change. Similar to an Agile way of working, working in **cross-functional teams** consisting of business analyst, developers, testers and operational staff is the way people should function (Velasquez et al., 2014). Debois (2011) indicates that many organizations do create small teams, however, they make the mistake of designing them functionally based on technology and not on a service or feature. When small teams are responsible for a service, all roles needed throughout the whole process are represented in the team. In this way the team is end-to-end responsible for their service or feature that has to be delivered. This cross-functional way of working creates also an opportunity for employees to learn from each other. **Cross-skilled people** are emerging due to this method and organizations will also hire other types of functions. It is more focused on people who are eager to learn and have competencies in more than only one field. This relates to the learning culture that has to be around in DevOps{Formatting Citation}. Encourage people to gain more knowledge and let them discover other tasks within the process. Reaching this **learning** culture within the team can be approached from two perspectives. On the one hand this can be facilitated by the management with offering team member enough possibilities for this, but it should also be in the complete mind-set of people. People should not have to think individually anymore, but have to understand how **sharing their knowledge** will increase the team success. Iden & Tessem (2011) have also appointed the lack of knowledge transfer as one of the problems between the interplay of development and operations. Therefore, appropriate knowledge management has to be maintained along the process and within the teams. Next to this, a culture of **experimenting without regret** has to emerge around employees, which also can be seen as a blameless culture (Davis & Daniels, 2015). Employees have to be able to experiment within their tasks in order to learn and to operate innovatively. The **willingness of people to cooperate** with this new way of working seems to be crucial. Without the cooperation of all people involved, it will be useless to implement a new method or way of working.

3.3.5. Information

Information is another aspect that can be taken into account with the implementation of DevOps. In order to create a focus on flow, **transparency of organizational information** involved is necessary. An environment is necessary in which anyone and everyone feels that they can speak their minds, and more importantly, contribute, as Swartout (2014) mentions. There should also be spoken in one language across the entire process as much as possible. Erich et al. (2014) states that “developers and operations should try to make their documentation understandable by both sides” (p. 13). This can be achieved with agreeing upon **standards in documentation and coding**. To increase the flow within the process, all necessary information on the process and product have to be **shared across** all the teams (Hüttermann, 2012). This includes the fact that people have to be authorized to access all relevant information. Effective monitoring of information will also support the collaboration between the departments (Lwakatare et al., 2015). Several monitoring tools are available nowadays to **monitor team’s progress and system performance**. This information can also be analyzed and used as feedback to improve the entire service or process.

3.3.6. Organizational Culture & Structures

The organizational structures and culture will also affect the implementation of DevOps. As DevOps can be seen as an extension of Agile, **a change in organizational structure towards an Agile way of working** is necessary to implement a DevOps way of working. Working in an Agile manner means in many large organizations that they implemented the Scrum process with sprints, backlogs and all

associated meetings such as planning, reviews and retrospectives. This Agile way of working requires also new roles within the organization, such as Scrum masters, product owners and possibly Agile coaches. **Contracts associated to these new roles** have to be changed as well. The contracts have to be composed with different structures. Next to that, the incentives of everybody involved in delivering software have to be aligned (Debois, 2011). Therefore, there are not individual incentives anymore on the delivering value, but only **incentives on the team effort** on a particular business stream or value (Hüttermann, 2012). In relation with this, **appropriate KPIs** have to be formulated for DevOps. A lot of people think that within an Agile environment the new KPI will become velocity, however, this is not measurable. The main performance indicator will become business value and the impact of a certain team or function on this value. In order to ensure a successful change, **top management have to support and facilitate process** within the organization. Without commitment of the management, the transformation to a DevOps way of working is very difficult. Beside all this, the overall DevOps culture have to be in place. Walls (2013) defined the DevOps culture as the **culture of collaboration with open communication, incentive and responsibility alignment, respect and trust**. This culture is very important to get commitment to the new way of working and also to get the monitoring of information and facts as clear and honest as possible (Swartout, 2014). Melorose et al. (2015) states that changing the business process will affect the culture, so when the right processes for DevOps are implemented, the DevOps culture will also raise (Melorose et al., 2015). It can be questioned whether this is actually true and therefore it is important to analyze this during the research.

3.3.7. Technology

DevOps is an organizational change, but it includes also many associated technology practices. Continuous integration and continuous delivery are discussed in the process category. Continuous delivery is extended with **Continuous Deployment** in the delivery pipeline. Continuous Deployment is defined as the ability to frequently and reliably put new releases into production, with as much automation as possible (Shahin, 2015). Continuous Deployment may not always be desired as other external factors are limiting the need for this as stated earlier in the literature review. The three “continuous” concepts are all three implying the adoption of automation. Therefore, **automated activities** are seen as an important practice of DevOps. For the activities should be thought of automated configuration, automated tests and automated releases, where automated releases can be an option according to the continuous deployment option. Next to automation, applying **Virtualization & Cloud-based infrastructure and applications** are related to DevOps. Currently technology is expanding in its abilities and more and more technical innovations are easily available for organizations. Different resources or activities are nowadays available as a service, such as IaaS, PaaS, SaaS, XaaS. This enables the possibility to make phases in the delivery process more efficient and faster. Within Agile this is already done by **focusing on minimal valuable product and extend with features**. Together with the new technologies, this is and can be done even more. Focusing on a minimal value product (MVP) is important to see what the basic requirements are and to see whether an early prototyped version is satisfying these requirements. With automating and thus standardizing many aspects, more attention can be put on innovation and new features. Next to this, new technologies are also creating a new approach in how to design. The ability to work in the cloud and creating virtual servers makes it possible to **design for failure** (Interview 3). This means not preventing failure, but designing resilient services that can survive failures and ensures its availability and reliability (Abbadi, 2011).

3.3.8. Change Management

Next to the categories on the practices that are related to a DevOps way of working, the change management to transform an organization towards DevOps is important. In this category the change aspects that are most important regarding DevOps are mentioned. Change management is always important, as also proven for BPR implementations for example. With mentioning the most important aspects regarding a DevOps change, the most relevant aspects are researched instead of discussing general change initiatives. First, **create understanding of Agile/DevOps way of working at the involved employees** in order to get them along in the change. Next to that, **training of employees in handling the DevOps way of working** is important. Changing the process is important, but when the people do not know how to handle this new way of working, the process will not be successful. Due to

the automation present in the process, it can be important to **make an overview of the automated process pipeline**. Everybody should know how the standardized and automated process works and is connected. To guide the change management a **special group or team** can be formulated. This group can be formed by people within the firm or by an external party. This can support the transformation towards the right direction. In order to do that, the **employees of both departments have to be treated equally**. Iden et al. (2011) states as the first problem of the interplay of the departments is that the operations department is not involved in the requirements specification. Next to that it is good to make **results on the performance visible**, as well as the improvements of the transformation. In the beginning the performance will not directly be improved, but it will increase over time. The team can also increase this when they are enabled to be self-organizing and act proactive with creating new ideas and solutions by themselves. This will make opportunities for improvements visible and will encourage a **continuous improvement** environment. Finally, **feedback sessions** will enable more improvements. Within Scrum there are already certain feedback moments on the sprint and the process itself, however, feedback on the process and performance during the change creates opportunities to improve the change management within the team.

3.4. Chapter Conclusion

The chapter answers the sub question: *“What type of change model for the implementation of DevOps should be used and how does a first design of the implementation model for a DevOps implementation look like?”*. Based on literature research the necessary information and insights are gathered to answer this question. Within the research DevOps is approached as a business process change. Therefore, different business process change models have been researched and used to develop the conceptual model (Figure 16).

The structure of the model is created from a combination of existing models of BPR and LSS as in that way both top-down and bottom-up approaches are included. The two categories environmental factors and performance measurements indicate the drivers of DevOps and how the performance is measured within large organizations. The six categories, process, people, information, organizational culture & structures, technology and change management within this model indicate the categories a change to DevOps is focused and what change management aspects are relevant to guide such change. An overview of all selected aspects is shown in Figure 18. The conceptual model is a theoretical design of the implementation model for DevOps. This creates a theoretical starting point for the focus points of the implementation of DevOps. In the next section of the Design & Development phase, the development of the model is continued. The value of this model will be researched based on case interviews with IT managers of large IT service organizations.

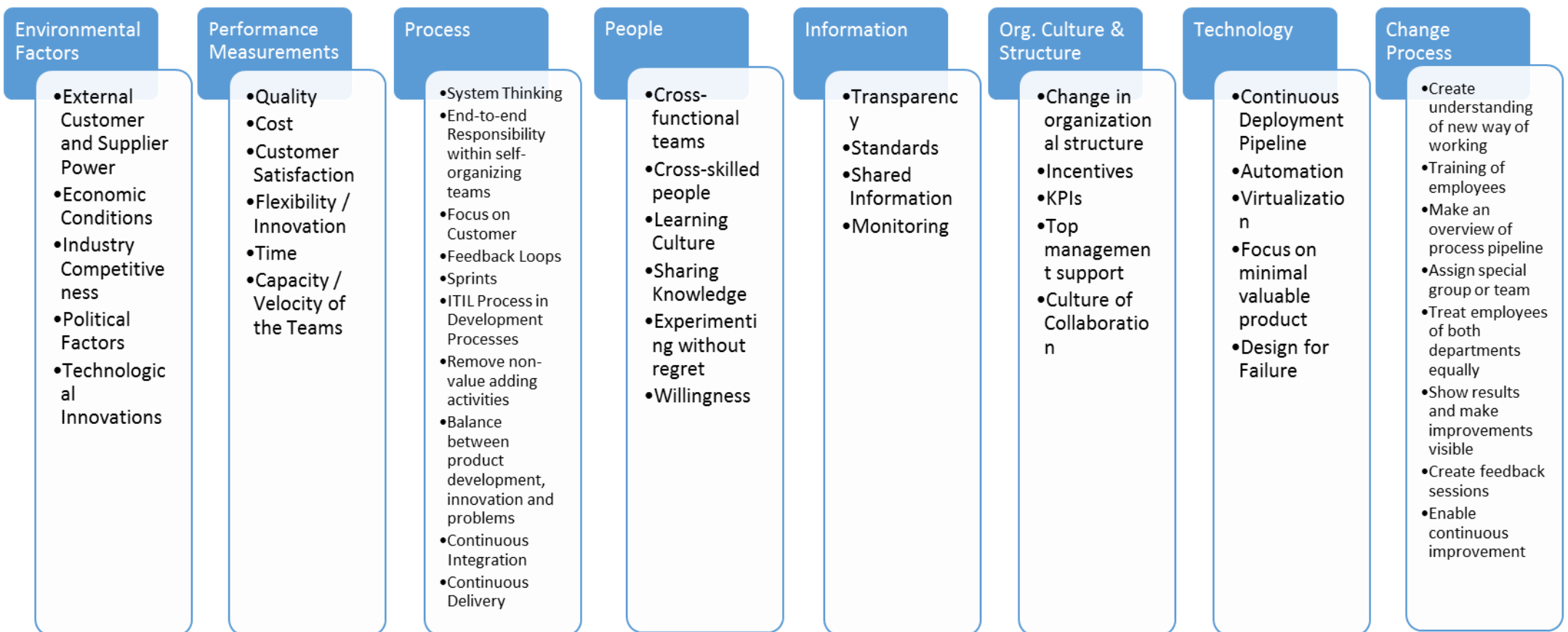


Figure 18: Overview of relevant DevOps aspects within the categories of the theoretical model

Chapter 4

Dvelopment of the
Implementation Model

4. Development of the Implementation Model

4.1. Chapter Introduction

This chapter provides the answer to the sub question “*What is the value of the theoretical design for large IT service organizations based on empirical research and expert knowledge?*”. This sub question refers to the next part of the Design & Development phase of the research methodology. Within this section the theoretical model that has been developed in the third chapter is being developed further with more practical research to increase the value of the model for large IT service organizations. Therefore, case interviews at large IT organizations are conducted. The additional findings during the cases are evaluated by experts in an evaluation session in order to state the value of these findings. After this section of the Design & Development phase, the theoretical and empirical conclusions will be used for the development of a final model.

First, the research methods are discussed in section 4.2. Section 4.3 provides information on the selection procedure which is used for the cases and expert selection and an overview of all cases. Section 4.4 provides an overview of the case interview protocol that has been used. Section 4.5 discusses all results and additional findings of the interviews and section 4.6 discusses the evaluation on the additional findings. Finally, conclusions on the results of the case interviews and evaluation session are drawn in section 4.7, which are the input for chapter 5 to develop the model.

4.2. Research Method

The research methods that are used within this section are case interviews and an expert evaluation. These methods support the development of the implementation model as with these methods the practical value of the theoretical model was assessed. This practical value helps to make the generic theoretical model specific for large IT service organizations. This is done by conducting case interviews at different large IT service organizations with IT managers or comparing roles. These can be seen as small case studies, where only one or two people of an organization are interviewed. Not entire case studies have been performed as this was not in line with the goal of developing a general implementation model for cross-industry organizations. Instead of two large case studies with many interviews per case for example, multiple cases have been involved by interviewing at least one person per case. The participating organizations are already working Agile and have the intention to move towards a more DevOps way of working. Case interviews with their IT managers create an insight on which elements are already implemented and on which element they would focus the implementation. During those interviews, additional findings aside of the model can be found. These findings are evaluated by an evaluation session with expert of Accenture. This is done to research the value of these additional findings as these findings were not included in the interview protocol. An evaluation session with experts is a suitable method to evaluate the reliability of such additional findings. The experts have more experience regarding the topic and can indicate whether this is a single statement or a common fact or event. The sections 4.3 and 4.4 provide more detailed information on the used research method and the procedures.

4.3. Case & Expert Selection

Due to the cooperation of Accenture with the research, clients of Accenture could be approached to participate as a case for the research. The clients that were approached are all IT service organizations for which DevOps could be beneficial. The selection criteria for the case organizations were based on providing an IT service, organization's size, current way of working, DevOps intention and availability. First, the organization had to offer an IT service and the size of the organization had to be large. The large size is within the research determined by the amount of employees, preferably over 1000 employees. Secondly, the current way of working had to be Agile within their teams. Accordingly, the intention or readiness had to exist to move towards DevOps in the near future. This was examined by people from Accenture who could indicate this based on their experience at certain organizations. During the cases, this was also questioned to the interviewee. Finally, the availability of the interviewees affected the possibility of conducting an interview at an organization. Selecting an interviewee within a particular organization is done based on the roles within the organization. As the scope of the research

was set on a model for the IT manager, all interviewees have such function or a similar role within these large organizations. The operating industry of an organization was not a criterion for the selection. The industries in which the organizations are operating differs as much as possible as the goal is to draw general conclusions on the implementation process of DevOps. Selecting organizations cross-industry prevents the research from drawing conclusions that are sector specific and will create a more general view over more industries.

Eventually, eight organizations have been involved in the research, which resulted in nine cases in total. To gain enough information on these cases, fourteen interviews are conducted. An overview of the cases and the specific roles of the interviewees is given in Table 3 and the case descriptions are included in Appendix E – Case Descriptions. The cases will be used anonymously in this rapport. First, each case has got an indication with a capital, ranging from A to I. Second, the operating industry of the particular organization is selected. For this selection, the five operating groups of Accenture are used, which are: Products, Financial Services, Communications & High Tech, Resources and Health & Public Services. The organization's sizes are indicated with the indicators Large, Medium and Small, based on their number of employees. At least one interview is done per case, but in some cases more interviews have been conducted. For each interview the current role of the interviewee is shown, as well as the amount of years of experience in the particular function.

Table 3: Overview of the characteristics of the cases

Case	Operating Group	Organization size*	Interview	Role	Years of experience**
A	Financial Services	Large	A1	IT manager / Chapter lead	6 years
B	Communications & High Tech	“Medium”	B1	IT manager / Agile coach (Accenture)	7 years
B	Communications & High Tech	“Medium”	B2	Chain manager	3 years
C	Communications & High Tech	Large	C1	Infrastructure consultant	10 years
C***	Communications & High Tech	Large	C2	Technology Consulting Manager (Accenture)	9 years
C***	Communications & High Tech	Large	C3	Technology Consulting Senior Manager (Accenture)	5 years
D	Products	Large	D1	IT manager Operations	9 years
D	Products	Large	D1	IT manager Development	3 years
E	Products	“Small”	E1	IT manager / Product Owner	4 years
F	Financial Services	Large	F1	IT manager	2 years
G	Financial Services	Large	G1	Tech. architect	1+ years
G	Financial Services	Large	G2	Tech. architect	1,5 years
H	Communications & High Tech	“Medium”	H1	Service manager	4 years
I	Products	Large	I1	Development manager	5 years

* Organization size (number of employees): <100: small; 100-1000: medium; >1000: large

** Years of experience in the particular function given

*** Interviews done to gather extra information on the case, not followed entire interview guide during those interviews

As can be noted in Table 3, in some cases the organization size is not indicated with ‘Large’. These cases are small exceptions on the formulated criterion where each organization size has to be preferably over 1000 employees. When looking for qualified organizations, the intention was to adopt only large

organizations. However, when the interviews were conducted at most of the organizations, some interviewees referred to a subsidiary organization of the large organization when talking about a more flexible way of working. Most of the time these subsidiary organizations are smaller and have a flexible culture as they are often relatively new. Some interviewees indicated that the adoption of a new way of working was started at those smaller subsidiaries. Therefore, it is interesting to take these smaller organizations also into account. To be more specific on each case, the relations between the cases is explained. Case B is a subsidiary organization of the organization used for case C. The interviewees of case D recommended to talk with another department of their organization, often seen as small enterprise within the whole organizations, which is case E. The organization of case H was involved through to the client connection with the organization of case C. This is not a large organization, but it was recommended by an Accenture expert to see how they have implemented an Agile way of working and their desired movement towards a more DevOps way of working.

After all interviews have been conducted and the data was analyzed, an evaluation session was held. During this session, most of the additional findings were evaluated. The evaluation session was part of an awareness session for DevOps. This session was held internal at Accenture and included Accenture employees within the Infrastructure department. Also some interested people of the Technology department participated. In total the session was participated by 17 Accenture experts. The selection of these experts was not done by the researcher, but based on an invitation for the session. The invitation was sent to the entire Infrastructure department within Accenture the Netherlands. The level of the experts is varying from analyst, consultant to (senior) managers.

In the session a presentation was held by an Agile/DevOps expert and during this presentation the researcher presented several statements and questions to the group on a large screen. In order to introduce DevOps and the research to the audience, some slides were shown and some example questions were asked to get everyone at the same starting point. The statements and questions that are used in the session are shown in Appendix D – Questions Evaluation Session. Everyone was able to provide an answer to the question or an opinion towards a statement via an online application, called “Mentimeter”. The possible answers to those statements were ‘true’ or ‘false’ and the summaries of all answers can be found in diagrams in Appendix G – Results Evaluation Session.

4.4. Case Interview Protocol

The next section provides more details on the procedures of the conducted interviews. By conducting these interviews, the categories of the conceptual model are reflected from a practical perspective. This creates a view on the model for large IT service organization which can help to develop the implementation more specific for such organizations. The section explains how primary data is gained through the interviews (4.3.1) and also what kind of secondary data is gathered (4.3.2). The drawbacks of the interviews are reflected in section 4.3.3.

4.4.1. Primary Data

The research data is retrieved by conducting interviews at different large organizations. In this way empirical data is retrieved with using qualitative research. The conceptual model of section 3.3 is through these interviews further developed towards a specific DevOps implementation model for large IT service organizations. The theoretical model gave a clear overview of the general business process change model adjusted to DevOps and with this practical data the model is more focused on only large IT service organizations.

Each interview took approximately an hour to two hours and was done in face to face meetings. All interviews were done in Dutch as all of the interviewees have this language as native language. The interviews were recorded after approval by the interviewee and after each interview a transcribed version was forwarded to the interviewee. In this way, the interviewee could evaluate the content on completeness and optional misstatements. After the approval of the interviewee, the document was used for qualitative analysis. Summarized conclusions and quotes are used in the results section (4.5). As all interviews were held in Dutch, these quotes are all in spoken language translated by the researcher as accurately as possible.

The interviews are semi-structured and the interview guide is presented in Appendix C – Interview Guide Case. Some questions were open and some questions were more closed with guiding the interviewee towards an indication of a value between high or low for certain aspects. First, the interviewer asked general questions, such as the current way of working and their definition of DevOps. After the general questions of the interview, the researcher introduced the research definition of DevOps. This definition is used throughout the rest of the interview to ensure a common starting point of all interviews. The research model is also shown to the interviewee with its structure and the categories of the model. Each category consists of several aspects that are explained earlier in chapter 3. Three questions per aspect have been asked: (1) Indicate whether this aspect is already applied in the current situation, (2) indicate the degree of each aspect at your organization and (3) indicate the importance of each aspect when implementing DevOps (Table 4). When a IT manager could not choose between two numbers for the degree, the researcher decided to select the higher or lower number based on the previous answer of ‘yes’ or ‘no’ respectively. The detailed results per category are shown in Appendix F – Results Detailed Tables. With calculating the difference between the degree and the importance, the so-called delta, the gap for each aspect is indicated. Based on the size of the deltas, the most important aspects to focus on are presented.

Table 4: Example of the interview guide per category

Category X

When implementing DevOps, which aspects would be/are the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the importance of each aspect when implementing DevOps:				
	Yes	No	1	2	3	4	5	1	2	3	4	5
Aspect X												

In order to gain more insights, the reason behind each indication was being asked. However, it turned out that almost all interviewees were already explaining many choices by themselves. Another question that was asked per category is whether they missed something on the list of aspects for that category. After discussing the model, the interviewer asked to identify the largest barriers for them to implement a DevOps way of working. Next to that, they were asked if they still had extra insights that they would like to share or any tips for the interviewer to read, watch or do regarding the research.

4.4.2. Secondary Data

The conducted interviews are seen as the primary data for this research. Besides this data, there is also secondary data retrieved. Most of the times it was recommended by the interviewees to look into this extra data. The secondary data is used to complement the cases with extra insights from earlier researches, projects, papers or books. For two cases an observation of the way of working of one day at the organization was done by the researcher where several meetings were attended. Relevant observations have been integrated in the results. Secondary data is not available for each case, only when it was available and useful to consult. In the case descriptions is indicated when secondary data is involved.

4.4.3. Drawbacks

The case interviews have other drawbacks that are necessary to mention already. First, an advantage is that all case interviews were done face to face. Nevertheless, small differences in the role and department of the interviewee can be seen as a drawback as this leads to another perspective. Due to time limitations and the availability of people, it was not always possible to talk to a person with the same role and department at all cases. Another reason for this is that roles and departments are different at each

organization, which makes it hard to identify whether people have the same tasks. However, the most corresponding role or function to IT manager has been interviewed at each case. Next to that, it was identified by an expert of Accenture (Interview 7) that it could be hard for the interviewees to indicate the importance of the aspects as managers have no reference. This is being prevented by stating that the interviewees should indicate whether they think it would be important for them in their organization or, when already further in the process of implementing DevOps, whether they focused on the aspect. It also has to be mentioned that the interviewees sometimes doubted between two numbers for the degree of implementation. In that case, the researcher decided to select the higher or lower option of the two numbers by looking at the Yes or No indicator respectively. Finally, a drawback is that not everyone has filled in the numbers completely. This limiting the insights on the aspects, however, when analyzing the results, it is taken into account by calculating the averages.

4.5. Results

During the interviews the research model of Appendix C – Interview Guide Case is used. This section provides a summarized overview of all results gained during the interviews: the values given in the model, the comments on the given value and the answers given to associated questions of all interviewees. First, the general results of the interview are presented in section 4.5.1. Secondly, in section 4.5.2 the results per each category of the model are discussed. Finally, additional findings are summed up in section 4.5.3. The additional findings are the result of the open questions during the entire interview, but also from interesting, recurring insights that are mentioned while answering other questions of the model. All quotes of the interviewees used in the text are translated by the researcher from Dutch to English. When a quote is presented, the associated case is indicated by its case capital and interview number.

4.5.1. General Results of the Interviews

The general answers provide insights on the view point of the interviewees and also what the current way of working of working at the organization is. This can contribute to the overall research as a final definition of DevOps does not exist. Knowing the perspective of the participating IT managers on DevOps contributes to the research as the research can be reflected upon this perspective. After this question, the research definition is introduced during the interviews and this definition is used throughout the rest of the interview. In order to use the answers and comments, quotes and summarized conclusions are used in the text. The values high, medium, low are used in different ways and per table will be explained how the researcher has interpreted these values.

4.5.1.1. DevOps

In the beginning of the interviews, the interviewees were asked what DevOps is in their eyes. It resulted in many different perspectives on DevOps.

“In my opinion DevOps does not exist. The name is only created to identify the merge of the two departments development and operations as previously there was a distance between them. Within DevOps you are one team and you build and manage your application or multiple applications, how you do it does not matter, you just do it.” (A1)

“DevOps is for me, not in the traditional sense of the word but how it is intended, it is a collaboration between operation and innovation, with the serving the company with the same purpose. This is sometimes where it goes wrong, in the interpretation, how it is used and implemented.” (C1)

“We approach DevOps and Continuous Delivery separately from each other. Continuous Delivery is all technology that is needed to go to production without long processes. DevOps is more the culture of collaboration and the people side of the process. There are models that state DevOps is something to accomplish CD or the other way around. That is a bit ambiguous and that is why we take it separately. We argue that you could do DevOps, without CD. I do not know whether you can do CD without DevOps.” (D1)

“At the same time, you see that there are needs within a team in which a shared layer exists. DevOps and CD is overlapping. You can do CD with all kind of tooling and technology, but you also want a certain mind-set and feedback loops which is more DevOps again. It is closely intertwined, but we choose to make it two initiatives in order to give both the focus it needs and not forgotten.” (D2)

“We call ourselves a “blended team”, because each team member has multiple skills. The developers are full stack developers (DevOps), they also maintain their own applications/products. We develop APIs and do everything in the Cloud. Within the team we rotate tasks regularly in order to keep track of each other’s work.” (E1)

“DevOps is a way of working, but supported by much technology. It is a shift, mainly a mind shift, that is visible in the way of working, Agile culture, continuous improvement aspects, within QA (working with KPIs), but an important technical component is also changing. I think it is characteristic that it is going throughout the whole company, not only within the ICT department or only at the business or at the floor. It goes top down, from left to right through the whole company.” (G1)

“It is a way of working. It is not pure technical, although you need technical support to make it successful. It is a way of working, in which the team is completely responsible for the whole process of the product they are working on, so as well as the early design to the production and also the operational support that is needed after that. This creates direct responsibility of a chain, where this is now often thrown over the wall and that is thus the biggest problem. It actually ends as the information system is abolished and you will start a new DevOps team.” (G2)

“DevOps is a team in which multiple disciplines are present, not in function, but in role in which blurring of roles is not a problem. The team is responsible for creating and reducing nice solutions and taking care of stability when in operations. That is the one-liner associated to DevOps. There is much around it and the question is how you are going to do that; how does your application landscape look like? When you have many small servers, it is different than when you have one big monolithic system. You want to get rid of the monolithic system, but that takes time and you need a route to peel this. There is much culture in it. Development is responsible for creating new business value, while operations does not want errors.” (I1)

The different perspectives of the interviewees on DevOps confirm that there is not one clear definition of what DevOps exactly is and which practices it covers. Nevertheless, recurring elements can be found within the different definitions. The aspects culture, technology, information, people, process, change approach are mentioned multiple times. It is also acknowledged that DevOps is about merging development and operations with multi-disciplined people, system thinking with collaboration towards the same purpose, applied throughout the whole organization. These elements match the four DevOps value that are defined by theory being: systems thinking, focus on flow, amplify feedback loops and culture of continual experimentation and learning. Therefore, the starting point of defining DevOps as a cultural movement with a number of practices regarding people, process and technology is suitable. This justly relates also to the choice of a business process change model and the categories that are defined in the model fits with DevOps.

After this topic, the interviewees were introduced with the research definition of DevOps and the model in the interview guide. Therefore, from this point the interview took the formulated research definition as basis.

4.5.1.2. Focus areas

During the interviews, the first question after introducing the research definition of DevOps and the model was about the areas the managers would focus on when implementing DevOps. For this question the five categories of the model were showed to the interviewee, without explaining the research model yet. The results are shown in Figure 19.

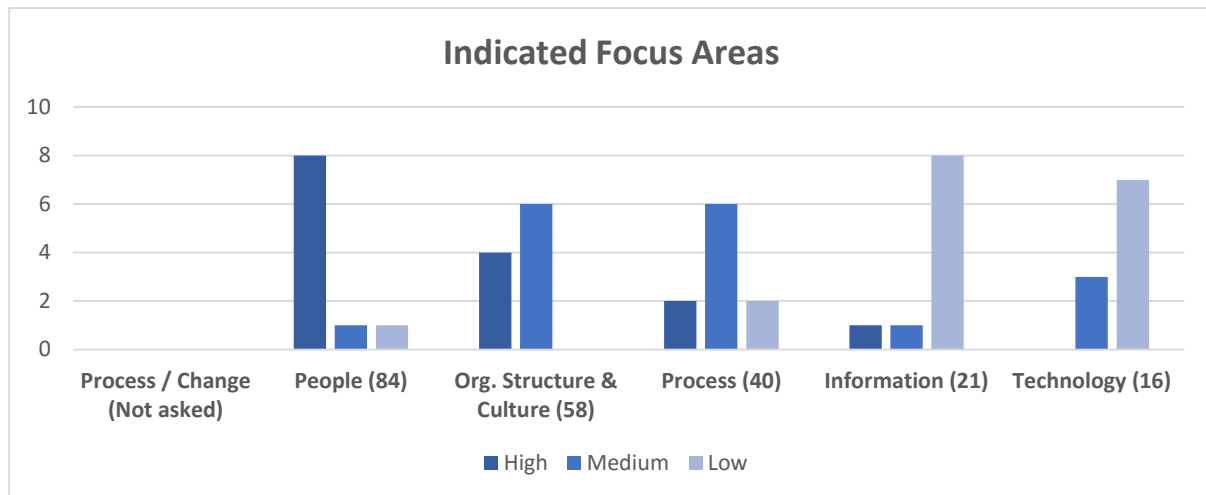


Figure 19: Overview of the Focus Areas

Interviewees were asked to indicate the factors on which they think a DevOps implementation should focus on with using a scale of high, medium and low. As can be seen in the table, these results are shown by the sum of each scale per category. The categories are also ranked based assigning a numeric value to the scales (High = 10, Medium = 3, Low = 1). The total sum of these outcomes are shown behind the category names and shown in the ranked order from high to low which is in that case from most important to least important.

First of all, it has to be mentioned that most managers started their answer on this question with stating **all areas** are important and also interdependent. “*All areas are important and even very important*” as a manager argued, “*but you may have to choose a first starting point to focus on*” (A1). Accenture experts were indicating that organizations are mainly focusing on the technical area when talking about DevOps. Nevertheless, many managers were arguing the technology part should be of least concern and the implementation process should not start by focusing on implementing new technology. Even a technical architect’s opinion was “*to focus on the people before paying attention to technology*” (G1). **Technology** and **information** are often seen as a mean and facilitator for the implementation of DevOps, not the main focus points: “*the design for information has to be adjusted, but that is less important, it is more a tool. Also technology will not be changed in high degree for your application, but you will add extra tooling*” (H1). Another technical architect states that the technology is actually the easiest step as “*integrating the technical ‘building blocks’ is hard, but much standard tooling is nowadays available for such integration*” (G2).

Almost all managers are indicating that the **people** within the team and organization are the most important to focus on, as one manager states: “*You have to start with the people, the people matters the most*” (E1, F1) and another: “*When the people know where it comes from and what it means, the rest will follow much easier*” (G1) and another one emphasizes: “*You have to take the people along*” (C1, G2). Many managers are acknowledging the importance of an appropriate mindset, which seems to be a proactive and cooperative mind-set. A manager sees the mind-set as a critical factor: “*The mind-set has to be there, otherwise it will definitely fail*” (F1).

Next to the importance of people, the **organizational culture & structure** that is present at an organization is indicated as highly important. A manager phrases this importance very clear: “*If you hold on to the silos within your organization, the silos will always counteract your effort of implementing a DevOps way of working. A silo will keep on being a silo with each their own interests and KPIs, so despite a similar mind-set, the silos will never work towards a common goal.*” (F1). The right culture will be created by the people according to most of the managers: “*By focusing on the people, the culture will eventually also change*” (G1, I1). However, the corresponding organizational structure should also be there, according to the next statement: “*You may want to change the culture, but the management structure has also a determining effect. When for example a desire exists on the floor to work in cross-functional teams, the management structure with contrasting contracts is making this difficult.*” (I1).

Also the **process** is given a high value on the importance, moreover *“the process that is going to be handled within the team and organization”* (E1). A manager said: *“When the right mind-set is there, it is important to design the organizational culture & structure and the process accordingly”* (E1, F1). Managers mention the organizational culture & structure and the process interchangeable in order of each other. A manager states *“the people have to be taken along and the processes have to be adapted to this, it is not right to keep current processes and adapt new technology to those processes”* (G2) and another opinion is that *“the interpretation of the processes will come after the organizational culture & structure are widely shared”* (A1). One manager acknowledges that they had to rethink the business processes as their entire process needed a different approach with other meetings and objectives (H1). The current processes within the large organization are also encountered as a difficulty to implement a new way of working and therefore attention should be focused on that (I1).

All these scales and comments were given in the beginning of the interview, before showing the research model to the interviewee. During the rest of the interview, each category was discussed in more detail. It is analyzed whether the first ranking matches the answers given during the rest of the interview as an extra validation. In order to do this, the average ranking score is calculated based on the answers provided by the interviewees. It has to be noted that not all interviewees gave a score for each aspect. However, by taking this limitation into account during the calculations, the average is based on the total amount of scores given specifically per aspect. With these aspect averages the total average per category is calculated. The results of this is shown in Figure 20. The figure shows one more category that was not taken into account within the focus areas. This category is the overall change management that is associated with the DevOps implementation within organizations. This is indicated as the most important for a successful implementation of DevOps. Beside this category, the table looks quite similar to the earlier table. The people factor is scoring the highest comparing with the other four categories. Process is indicated higher than organizational culture & structure, but from the different comments of the managers it was already noticed that these two are interchangeable. The same seems to apply to the categories technology and information, as both of them are often seen as a mean to facilitate the implementation of DevOps. The results validate the given answer to the first question.

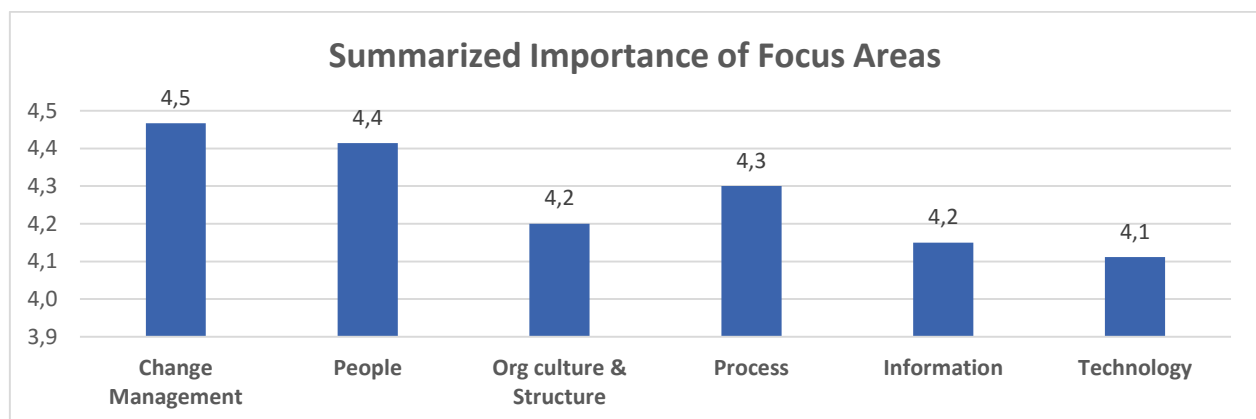


Figure 20: Importance of the focus areas

Figure 21 shows how managers indicate their degree of those categories within their own organization. The maturity of people and technology is the lowest, while the rest is around the same level. Combining the two figures lead to the conclusion that the people category is crucial, together with the process change category. It also validates that associated technology is less implemented, but also less important for organizations to focus on in the first place.

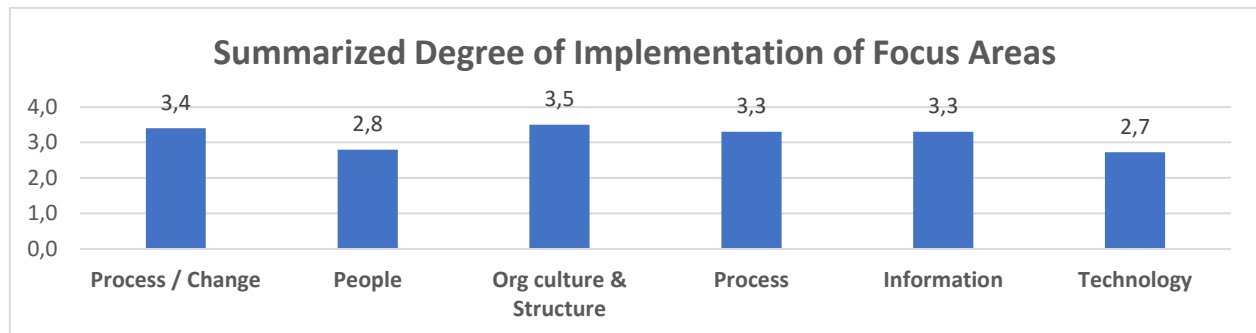


Figure 21: The degree of the focus areas

In conclusion, all theory-based areas seem to be important for large IT service organizations regarding DevOps. A ranking with small differences can be made by placing people on the first place, together with the change management. As second focus the process and organizational culture and structures is appointed, where after the information and technology categories are named in the ranking. This ranking confirms again that DevOps is mainly a cultural movement, but also that changing a culture and people is the hardest part, where changing technology seems easier. The extra confirmation of the research definition and the categories contributes to the development model. The ranking of the focus areas contributes to developing the model more specific for large IT service organizations.

4.5.1.3. Environmental factors & performance measurements

The two categories ‘environmental factors’ and ‘performance measurements’ were discussed shortly in the beginning of the interview. This was done together with explaining the research model in more detail to the interviewee. The overviews of the results on the two categories are shown in Figure 22 and Figure 23.

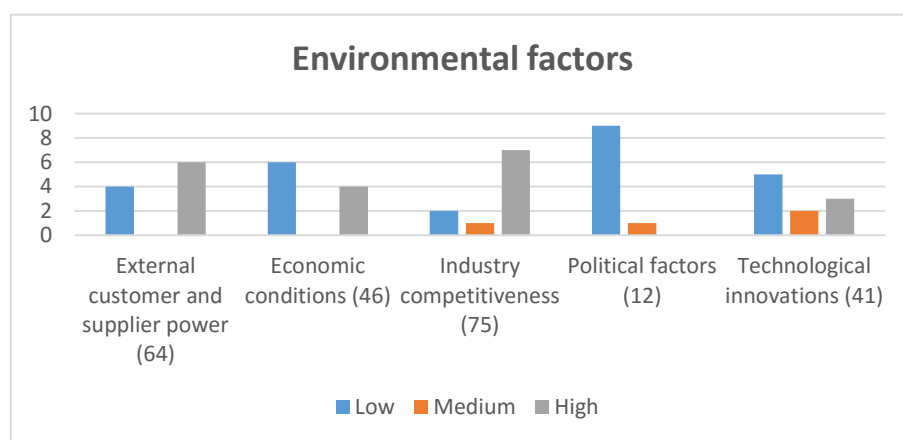


Figure 22: Graph with overview of answers on environmental factors

The factors industry competitiveness and the external customer and supplier power are seen as the most important drivers for implementing DevOps, where many managers were stating that this is only because of the external customer power. Suppliers have less influence on their process, moreover, it is often the other way around. It is surprising to see that the factor ‘technological innovations’ is not scoring very high, as DevOps prescribes the use of many innovative technology and tools to support the DevOps Way of working.

Within performance measurements the aspects customer satisfaction, flexibility and innovation are high. IT managers are acknowledging that the measurements have to be changed within the DevOps Way of working. This aligns with the Agile approach instead of the waterfall approach in which the fixed and variable metrics are already changed (Figure 17). Therefore, the aspects time and cost are not indicated as important measurements for Agile and DevOps anymore. Aside from the mentioned aspects, most

managers indicated that the aspect value becomes most important, although also most of them acknowledge that they do not know how to measure that.

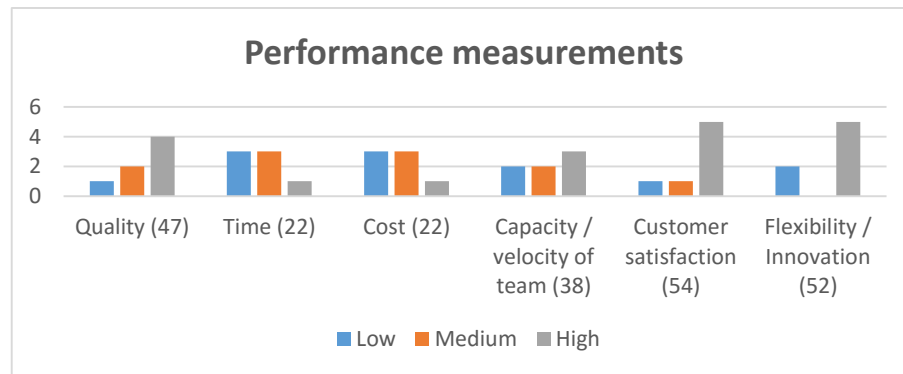


Figure 23: Graph with overview of all answers on Performance Measurements

4.5.1.4. Sub conclusions

The previous sections have described all general results. Based on those results, it is possible to draw some sub conclusions.

- DevOps has no common definition and can be viewed from multiple perspective. However, there is a common agreement on the importance of the cultural aspect of DevOps. The other elements that are mentioned related to the theoretical-based values of DevOps: systems thinking, focus on flow, amplify feedback loops and culture of continual experimentation and learning system thinking. The involvement of technology is also indicated within the definitions.
- All identified areas are important for DevOps, although the focus for the implementation should be on people as IT managers argue that people and culture are the hardest to change and indicate that the right mindset is not yet present within their organization. The technology requires also attention, but this aspect is perceived as a relatively easy change when the people have the right mind and top management supports the DevOps implementation.
- Due to the competitive environment and the external customer, large IT service organizations are driven to implement DevOps. In relation with that, value is the most important measurement to measure the performance of the business.

4.5.2. Results per Category of the Model

This section presents the results related to the aspects per category of the model. The results per each category contribute to the model in multiple ways.

First, by complementing the categories of the theoretical model with empirical data at large IT service organizations, the model is developed into a model that is specific for large IT service organizations. The qualitative answers of the IT managers will be analyzed and based on their comments, conclusions will be drawn. Next to that, the degree of implementation and importance per aspect is questioned during each case interview. The indication whether organizations have implemented certain aspects and in which degree gives a more detailed insight on the current way of working at the organization. It creates insight on the DevOps aspects that may have been implemented already due to the Agile way of working within the large IT service organizations. Moreover, the importance per aspect provides insight on the most important aspects for an IT manager of a large IT service organization. Based on this value the priority can be indicated on which aspect the IT manager would focus first. In relation with the degree of current implementation, it can also be seen on which aspects the managers pay less attention due to the fact that they have already a certain level of that aspect.

The sections 4.5.2.1 to 4.5.2.6 will summarize the results per category by analyzing and showing the most relevant qualitative comments that are given by the IT managers. The order in which the aspects are discussed per category is based on the gaps between degree of implementation and importance. By

comparing the degree of implementation and the importance, the delta between those values can be calculated. This can help to indicate the relevant aspect of DevOps within large IT service organizations. A large delta means that the degree in which IT managers see that aspect implemented in their organizations is small, although they value that aspect as important to implement. The larger this gap is, the more important it is to focus on this aspect. Smaller deltas can indicate that this aspect is already implemented at a suitable level according to the managers. When the delta becomes lower than zero, the aspect is implemented on such a level that the managers do not think it is important to focus on this anymore. In this way the most important aspects for the IT managers in larger organizations can be concluded. The detailed results per category are shown in Appendix F – Results Detailed Tables.

4.5.2.1. Process

When looking to the deltas for the category process, the top three of largest gaps is easily found (Figure 24). First, **continuous integration** and **continuous delivery** have the largest gaps in this category, where the gap of continuous integration is the highest of the two. However, within this research continuous delivery is defined as the next step upon continuous integration. Arguments show that most of organizations do continuous integration on a small scale in specific departments, but definitely not everywhere. Some organizations state that they do not have it yet or that some departments are between CI and CD, for example:

“As department we want to go to Continuous Operations and we are currently between continuous integration and continuous delivery. It is technically still not fully realized.” (A1)

“No, we are not doing CI but I do think it is a core characteristic of DevOps.” (C1)

“Yes, we are doing Continuous Integration, but it can be better. We have it until test, until test we can deliver everything with one push on the button.” (F1)

“Our focus point is now continuous delivery, but this includes also many other things, such as a big part of technology that we do not have in place yet.” (G1)

“We do partly continuous integration, but not yet Continuous Delivery. This would be a next step, together with extending continuous integration.” (H1)

“We are in the phase to go from CI to CD. We are able to do it, but we are not allowed to go over the wall to operations due to several contracts” (I1)

These statements confirm that continuous integration is followed by continuous delivery and the focus should be first on continuous integration. One manager, working as technical architect at the infrastructure side, argues that it differs per department whether it is necessary to use CI: *“For us this is less applicable, more for software development. Everything that we do demands new installations, which have generally a larger impact than application changes. Developers do use this, they are able to deploy their code on infrastructure, without really touching the infrastructure.” (G2).*

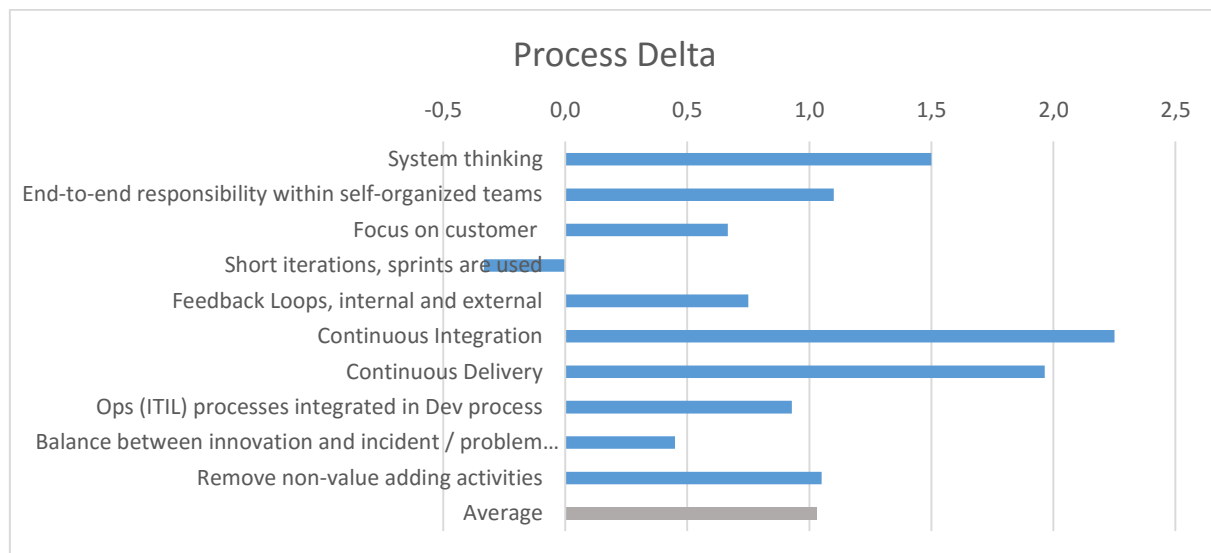


Figure 24: Graph with all deltas for the aspects within the category "Process"

The third largest gap is the aspect **system thinking**. Many managers state that the focus on the whole chain is important and that everyone should work towards a common goal. Managers are indicating this importance in the following quotes as: *"Focus on the chain. With the product owner, we are looking to the whole pipeline. It is important to focus on the chain and not to focus only on separate parts of it."* (B1), *"This is just very important; it is the whole new mindset that is associated with the culture."* (G2) and *"We don't have separate swim lanes in which we operate, we work towards our shared objective"* (I1). This is sometimes realized by managers responsible for a whole chain within organizations. However, the level of actual implementation at the organizations is not yet high enough. Most of the time this is caused by the silos in the organizational structure as two managers are arguing: *"We are too much in silos to do this"* (C1, G1).

Another manager puts an interesting note to this aspect by worrying that *"this aspect is important for Agile, but it may become complicated for DevOps due to all separate application teams with operations people included. How are all these applications then supported as there is not a large overall operations team anymore?"* (F1). This is a fair question and often asked by people. The reason for this is that there is not one specific organizational structure or model for DevOps. Therefore, people are not sure how the organization is going to look like with DevOps totally implemented. However, within different frameworks and organizational models this problem is solved in different ways, either it is by keeping a level one operational department or either with interconnecting all teams with certain overlapping chapters or interest groups.

After the three aspects with the largest gaps, the differences of gap sizes become smaller. **End-to-end responsibility within self-organized teams** and **remove non-value adding activities** have similar gaps. Self-organized teams are most of the time only organized within development or operations, not across both departments. Conflicting interests do still exist due to that reason, however, managers often try to improve this in an informal way. Autonomy (A1) and responsibility (G2) is very important according to two managers. However, *"teams have to get enough freedom to move, but they are not allowed to do whatever they want"* (G1). Most managers state that **remove non-value adding activities** is important throughout the whole business and not particular or DevOps. They do mention that the mindset of continuous improvement at people could be improved, but other aspects have more influence on this.

The same counts for **feedback loops**. Most managers state that they do have feedback sessions, also related to the meetings that Scrum describes, although this could be improved incrementally (F1). One manager argues that *"this is still done within one department and it would be better to communicate feedback throughout all departments in order to get the bigger picture more clear and to reach common interests"* (G2). **Focus on the customer** has to be focus on value, which is very important for DevOps: *"This value can be a direct value for the customer, but also an internal change that indirect affects the*

business value". (B1) Together with this, "you have to keep in mind what the organization's value and capabilities are, not only what the business wants" (G1).

All of the participating organizations use currently ITIL, fully or partly, within their operational phase. With DevOps, **ITIL will be integrated within the development processes**, however, not all aspects of ITIL will be used anymore within DevOps (B1). In order to do that, "both Dev and Ops departments were first working in an Agile way as much as possible. Eventually, ITIL processes will be adjusted and integrated in DevOps processes, as the two departments had to be integrated." (H1). Most managers state this less important as the ITIL processes will be adjusted, but they do state that process aligning is important. The **balance between problem development, innovation and incident / problem management** is related to this. As Dev and Ops are in most of the cases still two departments, the balance is not completely in place yet and therefore also less important as first the two departments have to be brought more together. Many development teams have reserved capacity on their sprint backlog for problem management. There is often still a conflict in priorities between developing new features and problems from operations, therefore common knowledge over both departments is important (A1). The interviewee of case H provides a nice example to show the importance of a right balance:

"We have reserved standard a percentage for old problems on the backlog and after care. At a certain moment we had no agreements on how much problems we would solve in that sprint, although we monitored each problem, we did not solve any of them. Until the stock of problems became so big, that it was not workable anymore and agreements had to be made. Due to the clear monitoring, we had good insights on the pile of defects. This made us realize that we could not only keep on building new things, but also had to pay attention to problems in the things we have built. Good agreements are therefore necessary and very important." (H1)

The aspect **short iterations, sprints are used** has a negative gap. The aspect is already implemented at a high degree at the organizations and due to this, the managers perceive this as less important for DevOps. This can be clarified as this aspect is very much related to the Agile way of working of Scrum that is highly used at the participating organizations. This is also concluded from the comments as most of the managers said that they adhere to sprints of two weeks. There is one manager who remarks that people have to keep in mind why sprints are used as "it should not be the case that you are obligatory to work in two weekly cycles, but you have to be able to argue why it has to be done so quick." (G2).

To summarize this category, Figure 25 gives an overview of the conclusions.

Relevant for DevOps	Already Implemented through Agile	Not Specific for DevOps
<ul style="list-style-type: none"> • Continuous Integration • Continuous Delivery • System Thinking • End-to-end Responsibility within self-organizing teams • Focus on Value • ITIL Process in Development Processes • Balance between product development, innovation and problems 	<ul style="list-style-type: none"> • Short iterations, sprints are used • Feedback Loops 	<ul style="list-style-type: none"> • Remove non-value adding activities

Figure 25: Overview of the conclusions within the category "Process"

4.5.2.2. People

The deltas of the category "People" show also interesting variations (Figure 26). The aspect that rises above the others as only one is **Experimenting without regret**. This aspect is indicated as highly

important by the managers, but the current degree of implementation is not yet very high as for example one manager mentions: *“This is not yet happening, but should definitely be happening”* (B1). Another manager states: *“Although I do not judge people on their mistakes, it is not embedded in the mind-set or way of working. It is however very important, as you can drive innovation with this approach”* (F1). However, there are also some departments where certain experimental characteristics are implemented: *“We have a beta version of our application, which helps us to test new creative ideas and new technologies”* (E1) and *“We have specific divisions that focus on experimentation, but that is outside the DevOps domain”* (G1). Next to that, some managers are indicating that people have to get the space and freedom to enable an experimenting culture (G2, I1). **Learning** is related to this experimenting culture. Within organizations where departments have already implemented DevOps in a certain level, learning is encouraged through for example awareness sessions (A1) and rotating tasks (B1, I1). Putting people within one team is also encouraging the learning as a manager explains: *“Where in the past new ideas were stopped by another department, this is now solved by putting everyone in one team and let them learn from each other”* (A1). The **willingness to cooperate** is already present at a certain degree most of the time, however, it can always be improved and it is very important that people are going along with the DevOps culture or change (A1, B1, E1, F1, G1, H1, I1). One manager acknowledges also the fact that *“it is important to focus on this aspect and you have to maintain this or rather, continue to focus on it”* (H1).

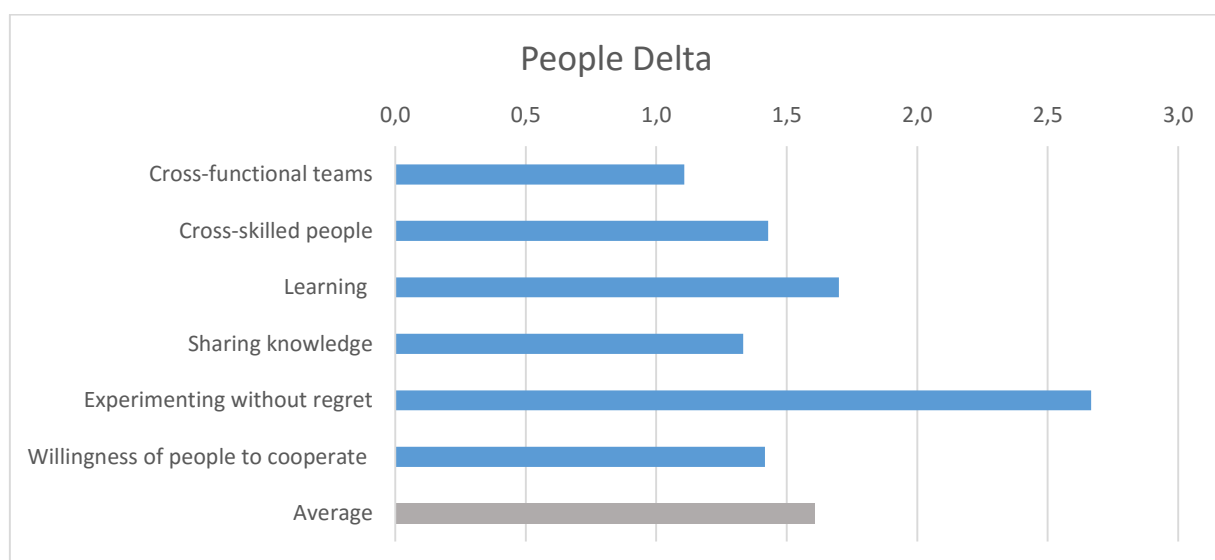


Figure 26: Graph with all deltas for the aspects within the category "People"

The aspects cross-skilled people, sharing knowledge and cross-functional teams are showing the smallest gaps. The fact that the gap of cross-skilled people is larger than cross-functional teams suits with the definition of DevOps. Cross-functional teams are common in an Agile way of working and cross-skilled people is seen as next step upon this. Agile encourage this already, although DevOps extends this aspect and sees it as requirement. Many participating organizations have **cross-functional teams**, although sometimes in an informal way, as this is also appropriate within a more Agile environment, only the operational department is not yet included. Often the developers and testers are working together within one team, also together with QA, although from an organizational structure point of view the teams are still silos. **Cross-skilled people** is not scoring high in the current implementation as the teams are also not yet fully cross-functional. However, almost all managers agree that within the DevOps mind set, all people are required to know the basics of everything, but will still remain a hero in their background skills. At a department which is working already according to a DevOps mind set, one moment an operational employee is a developer and the other way around (A1). **Sharing knowledge** is often done informally on-the-job for knowledge on small, daily tasks within the teams. For knowledge between different teams, some organizations use a knowledge base (H1) knowledge transfer sessions (B1), interest groups (G1) or so-called chapters (A1) for this. One manager is saying that knowledge management can be better as there is not a focus on documenting and

everything is changing quickly (A1). It is a challenge to keep the knowledge to the organization and not only within people.

To summarize this category, Figure 27 gives an overview of the conclusions.

Relevant for DevOps	Already Implemented through Agile	Not Specific for DevOps
<ul style="list-style-type: none"> • Experimenting without regret • Learning Culture • Willingness • Cross-skilled people • Sharing Knowledge 	<ul style="list-style-type: none"> • Cross-functional teams 	

Figure 27: Overview of conclusions within the category "People"

4.5.2.3. Information

The category "Information" contains four aspects. The gaps of the aspect **transparency** of organizational information and **shared information among the teams** are the largest (Figure 28). Transparency of information seems to be hard with working in silos. Managers identify that within a team information is being shared, but on organizational level it is limited. A very large information database is sometimes used for this matter, although the managers indicated often that transparency on the performance and process can be improved. In contrast, it is indicated by one manager that not all information needs to be shared with everyone. However, when there is a DevOps culture, with trust and shared purposes, this will be irrelevant. Lack of transparency between Dev and Ops is also still appointed as a problem, as these teams are not working together yet within Agile.

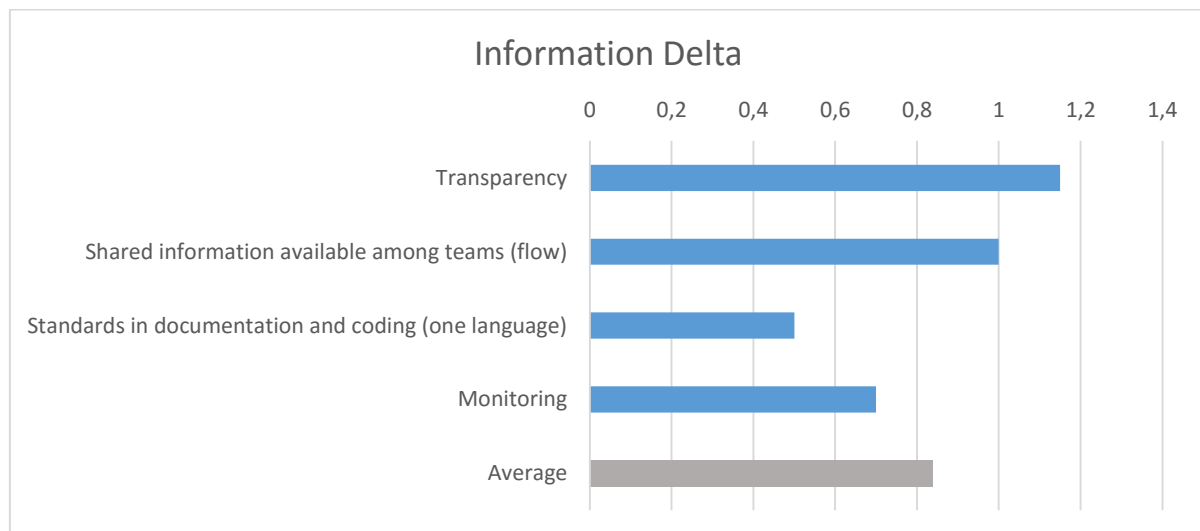


Figure 28: Graph with all deltas for the aspects within the category "Information"

Monitoring on the service performance is at some organizations being done by large dashboards or screens that hang around on the floor, where others for example desire to have such dashboards (H1). Automatic process monitoring within the technology is sometimes also done, which can be useful for the performance monitoring as well. Another organization has "special Agile coaches who take care of the process monitoring, next to the manager himself and weekly meetings in which the team discusses the process" (I1). Agreements on **standards for coding** is often already done due to an Agile way of working within the development teams. Moreover, documentation has already a lower priority according to the Agile manifesto (Beck, 2001). Speaking in one language is often also tried, although this is harder

with the different silos. There is often not one programming language used, but multiple programming languages with standards and agreements in how to code (E1, G1, H1, I1). One manager is stating that “each team can decide upon their own standards to a certain extent and within the chapter the overlapping standards are discussed” (A1).

To summarize this category, Figure 29 gives an overview of the conclusions.

Relevant for DevOps	Already Implemented through Agile	Not Specific for DevOps
<ul style="list-style-type: none"> • Transparency of organizational information • Shared Information among teams • Monitoring 	<ul style="list-style-type: none"> • Standards for Coding 	

Figure 29: Overview of the conclusions within the category "Information"

4.5.2.4. Organizational culture & structure

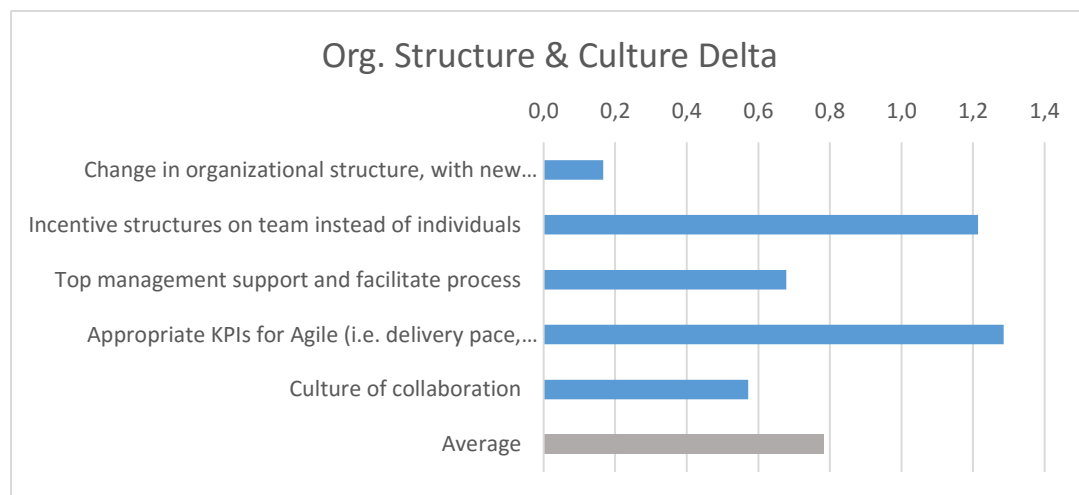


Figure 30: Graph with all deltas for the aspects within the category "Org. Culture & Structure"

Within this category several aspects regarding governance, people and structure are mentioned. Figure 30 shows that the aspect regarding the **KPIs** is scoring the highest gap. It is noticed that there are some differences in how the interviewees approach KPIs. First, one manager that is working in a DevOps environment states that they do not have KPIs anymore: “It is all learning by doing, where the only KPI is not getting over the budget” (A1). Some managers state that they are still being judged within Agile on the velocity, although they realize that this is not correct and this should be changed in measuring the business value (C1, E1 H1, I1). This is something that most managers do acknowledge but do not have in place yet. They struggle with how to translate this value into real metrics. Someone also mentions the importance of measuring fact-driven and data-driven, but they are not doing it properly yet (I1). Another manager is stating that the KPIs should be changed along the process and the maturity of DevOps at an organization (B1). Related to these KPIs the **incentives on the team instead of individuals** is indicated with a large gap. Almost all managers are stating that individuals are still rewarded and not the entire team. With using Scrum tools it is exactly traceable which task is assigned and performed by which team member (A1). One manager states: “Within our team we measure performance individually, but outside the team we are judged as a team. Everyone has his own responsibilities within the team.” (E1) Most managers acknowledge again that measurements should be based more on team performance in order to create common objectives, however, they are still struggling how this type of structure should be designed. It is also said that “the self-organizing concept initiates also that you judge people differently. People will judge each other, that is totally different than before” (A1). **Top management**

supporting and facilitating the process is indicated as already implemented to a certain degree, however, it can be improved at almost all organizations. It is important to remove the distance between the business and IT. Most of the time managers state that top management support a new way of working, but gives the freedom to middle management to interpret this new way of working in more detail. This is often perceived as a positive approach although they argue that the support could be more visible at certain moments. **Changing the organizational structure** is varying per organization. It was discovered that an Agile way of working already changed roles and structures. Different organizational models that embrace an Agile or DevOps way of working are used within the cases and other organizations have only changed some roles for an Agile way of working. For DevOps, the organizational structure has to be changed even more. Within the cases it is often seen that the organizations choose for a chain structure with so-called chain managers. Also the organizational models of SAFe and Spotify are mentioned during the interviews. Within this research it is not said which organizational structure is the best to choose, however, organizations have to think carefully about which structure would be most appropriate for the organization. This can be done according to multiple decision factors, such as the current structure, vision of organizations and the area of implementation. The delta of this aspect is small, which can be clarified by the high degree of implementation as the importance is indicated as high. However, the high value of implementation is based on the organizational structure on team basis. The entire structure of an organization is often not changed. Most managers (A1, C1, H1, I1, E1) indicate that the **culture of collaboration** is already present at the organization or division where they are working. However, it can be questioned whether it is really present as their perspective could be judged as subjective. Also the degree in which this culture is present could be lower than is meant within DevOps.

To summarize this category, Figure 31 gives an overview of the conclusions.

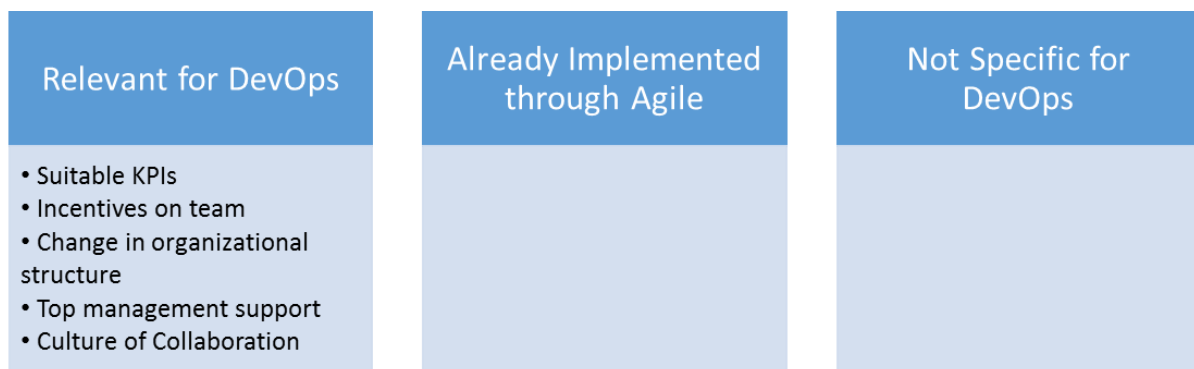


Figure 31: Overview of the conclusions within the category "Org. Culture & Structure"

4.5.2.5. Technology

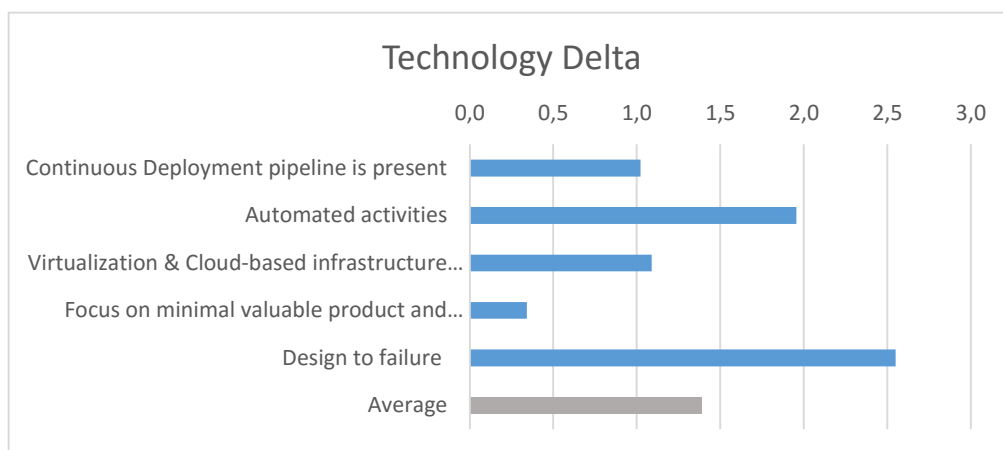


Figure 32: Graph with all deltas for the aspects within the category "Technology"

Within this category the ranking of the gaps is clearly shown in Figure 32. **Designing for failure** is scoring the largest gap, which is not surprising. This aspect is highly related to Cloud applications and these are not yet implemented at a large scale within the large organizations. Except from case E1, nobody has implemented this, although they all indicate designing for failure as high important. This also relates to the learning and experimental culture. One manager states that this is a nice to have and the other technology aspects are more important to focus on (G2). Another manager states that this relates also to the mindset that has to be at the people: “*You have to build something of which you also have think about it can be demolished*” (A1).

Continuous deployment is not desired as main goal for each organization, it depends also per division within an organization. First, the focus is on automating as much as possible. Most organizations have already undertaken certain steps towards **automation** and are still busy with extending this, also due to the fact that this is already important within the Agile way of working (A1, B1, C1, G1, H1). **Virtualization & Cloud-based infrastructure and applications** is also implemented to a certain extent already within the organizations. IaaS, PaaS and SaaS are often also present, although this is less applicable for the operational team. Cases E and H are suppliers of SaaS themselves. Another thing that is mentioned is that infrastructure is sometimes outsourced, which makes it hard to decide by themselves to implement the Cloud for example (H1). The **focus on minimal valuable product and extend this with features** is already created with working Agile. This makes the gap automatically the smallest.

To summarize this category, Figure 33 gives an overview of the conclusions.

Relevant for DevOps	Already Implemented through Agile	Not Specific for DevOps
<ul style="list-style-type: none"> • Automation • Virtualization • Design for Failure (Nice-to-have) • Continuous Deployment pipeline (Nice-to-have) 	<ul style="list-style-type: none"> • Focus on minimal valuable product 	

Figure 33: Overview of the conclusions within the category "Technology"

4.5.2.6. Change Management

The change management category is approached from how the change management is performed currently within the organization and which aspects the managers would rate as important within this process. During the interviews it was mentioned multiple times that all of these aspects are important, which can also be seen by the gaps. One manager summarizes it very clear: “Understanding, education, making an overview of what we are doing exactly, key group that keeps the focus and guides the process, treating everyone equally, show results, making improvements visible and share with everyone, keep everyone involved and improve continuously where it is possible, which you do based on feedback.” (H1).

In Figure 34 can be seen that three aspects are scoring the highest gaps. **Treating both departments equally** is very important within the change management. “*Often the operations department is not being seen as important as development is delivering new functionality, however, operations is actually the heart of the organization. I say sometimes: Just shut down operations for one day and you will see what is happening. This often creates awareness of how important operations is*” (C1). Due to conflicting interests of the departments this is hard to accomplish during the process, however, it can at the same time also be solved by removing the conflicting interests. A method to solve this problem suggested by

a manager is to let the managers of the two departments exchange roles for a while to experience the interest of the other department (C1). This helps to acknowledge the importance of the other department.

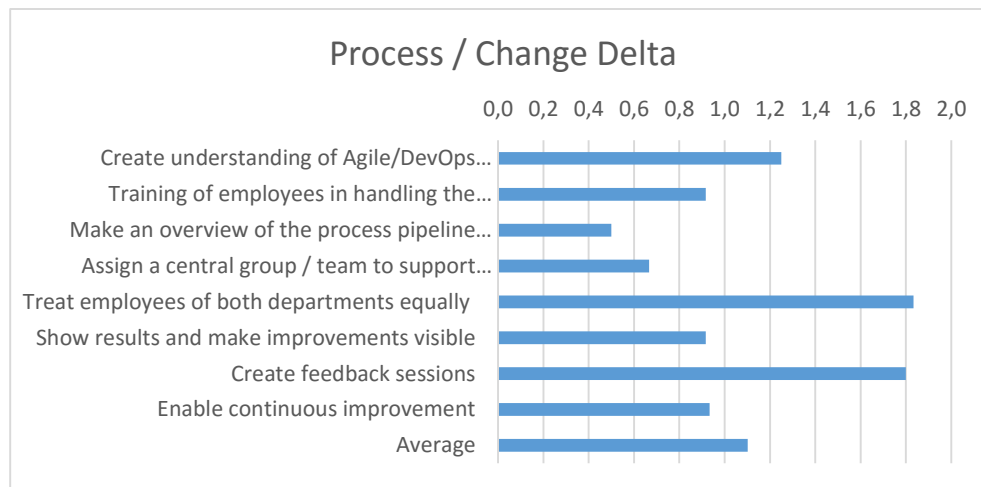


Figure 34: Graph with the overview of all deltas for the category "Change management"

This can also be improved by setting up **feedback sessions**, not only on the quality of the service, but on the ongoing change management that they experience. The managers are stating that this is already happening, but more focused on the feedback on the performance of the teams. Feedback sessions on the process could be improved, sometimes with help of Agile coaches that are involved at the organizations. This is related to the aspect **Show results and make improvements visible**. By organizing feedback sessions, the results on the change management and progress can be communicated. Sometimes monitoring is already used for this matter within the organizations. **Create an understanding of the Agile/DevOps way of working at employees** is indicated as important by the managers, as one manager states: *"When people do not see the urge of it, nobody feels like changing something"* (G2). Many managers are stating that this understanding is already present or upcoming at the involved people. However, the business has to be integrated with this understanding (I1). Some managers also indicated the **importance of working with tools or a process** instead of hearing it during a training for example (D1, E1, G2) The same opinion appears to be shared for **training employees** in handling the DevOps way of working: *"Training guarantees never that people are able to apply something, learning on-the-job is better. In that case people are working with it, understand it and eventually will become enthusiastic about it themselves."* (G2). **Enable continuous improvement** has to be an important drive within the change management. As one manager states: *"It is never finished"* (F1). **Assign central group or team to support transition to DevOps** is by most managers agreed (A1, F1, G1, G2, H1). One manager states that this did not exists, but agrees that an overarching coordinating team can be very useful (B1). This could also be a "virtual team" with people who align the process with each other. Another manager argues that the "just do it" mentality is necessary instead of a team. **Making an overview of the process pipeline** is indicated as less important as the focus should not be on documentation, but on people (H1). Some high level overviews on the process are made over time at most organizations, but it will always change (C1). However, a manager states: *"It is important to know what is automated and which building blocks exist. You have to know what you have in order to use it"* (G2).

To summarize this category, Figure 35 gives an overview of the conclusions.

Relevant for DevOps	Already Implemented through Agile	Not Specific for DevOps
<ul style="list-style-type: none"> • Treat employees of both departments equally • Create feedback sessions • Show results and make improvements visible • Create understanding of new way of working • Training of employees • Enable continuous improvement • Assign special group or team • Make an overview of process pipeline 		

Figure 35: Overview of the conclusions within the category "Change management"

4.5.2.7. Sub conclusions

The previous sections discussed the most important findings per category. In conclusion, the aspects are ranked on which are most relevant for organizations to focus on. Also the aspects that are nice-to-have, the aspects that are already implemented within an Agile way of working and the aspects that are not specific for DevOps within large IT service organizations are indicated per category. Table 5 provides an overview of all aspects, ranked according to the overviews within the results. The colors indicate certain characteristics of the aspects as can be seen in the legend associated with the table. This overview with its conclusions on the categories and their aspects is used to make the implementation model specific for large IT service organizations.

Table 5: Overview of all aspects per category ranked according to empirical data

Rank	Process	People	Information	Org. Culture & Structure	Technology	Change management
1	Continuous Integration	Experimenting without regret	Transparency of org. information	KPIs	Automation	Treat employees equally
2	Continuous Delivery	Encourage learning culture	Shared information	Incentives	Virtualization	Feedback sessions
3	Systems Thinking	Willingness	Monitoring	Change in organizational structure	Design for Failure	Create understanding
4	End-to-end responsibility within self-organizing teams	Cross-skilled people	Standards for coding	Top management support & commitment	Continuous Deployment	Continuous improvement
5	Focus on value	Sharing knowledge		Culture of collaboration	Focus on MVP	Show results
6	ITIL processes in Dev processes	Cross-functional teams				Training
7	Balance between dev, innovation and problems					Assign group

8	Sprints					Make an overview of automation
9	Feedback loops					
10	Remove non-value adding activities					

	Relevant for DevOps
	Implemented with Agile
	Nice-to-have
	Not specific for DevOps

Figure 36: Legend associated to Table 5: Overview of all aspects per category

4.5.3. Additional Findings during the Interviews

After discussing each aspect of a category it was asked whether the interviewee missed something on the list of aspects. The interviewer also asked what the most important barriers or limitations were for the organization to implement DevOps according to the interviewee. Another question was whether the interviewee had other tips or comments for the interviewer to take into account during the research. This resulted in extra information, some related to the model, others were new to the interviewer. Moreover, during the interviews certain things were mentioned that are not taken into account in the conceptual model. These things are interesting and important to mention as this makes the model more applicable for large IT service organizations. These additional findings can be categorized in two types, additional findings related to the categories of the theoretical model (4.5.3.1) and additional findings related to the implementation of a DevOps way of working (4.5.3.2).

4.5.3.1. Additional Findings related to the Aspects

Process

The conflicting interests of the departments have to be aligned. Operations is often used to a 24/7 culture where people are standby at all moments, where development is more designed according to a project and functional basis (C1). This relates mainly to the processes that are used. The development department is responsible for creating new business value and the operational department is focusing on minimizing disruptions of the system. This **conflicting interests** should be managed within the DevOps teams. During the process towards a DevOps way of working it is also very important to acknowledge this difference. One optional solution for this which was mentioned by a manager was to let the managers of the two departments switch from roles for a while, which will make them aware of the importance of the interest of the other department (C1). This manager is also stating that it is important to keep a level one operational department that keeps being responsible for the basic operational tasks.

People

Next to a cooperative and open mindset towards change among the employees, the factor of having a **proactive mindset and attitude** of the people in the teams is important. This relates to the category people. When a proactive mindset is achieved, people will also see what can be changed in their daily work. This will support continuous improvement throughout the whole process. Therefore, people have to think for themselves and should try to come up with solutions or improvements by themselves during their work. As one manager states that everyone should think in possibilities, instead of impossibilities: “When you do something over and over again in the same way, we expect that you will start thinking whether it can be done differently, more efficient. Thinking about the question ‘Why?’ at every level is

very important” (A1). Another manager states that people across all teams have to act proactive in showing their view or request, instead of complaining without doing something about an inefficient aspect (I1).

This seems to have influence on the HR model of an organization, which also relates to people. Multiple managers state that their organization is looking for another set of skills of their employees. People should have a **basic and broad understanding of multiple areas**. Heroes within only one aspect are being removed within organizations and this is in line with a DevOps values of this research. Such heroes can create a bottleneck in the system which decreases the flow in the process. Decreasing the influence of heroes can be reached with training the current employees towards a certain standard or by changing the recruitment criteria in order to hire a **different type of person, based on its skills set**. Also **commitment** of the people within the teams and other people that are involved is necessary, otherwise it will never be a success. As large organizations have many outsourced tasks to other stakeholders, it can be hard to get everyone on track and at the same page. This can be solved with multiple aspects within communication and agreements, but also in differing in the degree in which DevOps is implemented.

Organization Culture & Structure

The support of top management is also seen as a barrier by some managers. Some managers do acknowledge that the higher managers state that something has to be changed. However, there should also be given **budget, time and room to move** towards this new way of working as this is not done all by itself. In order to achieve this, it is important that the business sees the long term benefits of applying DevOps as on the short term it will only cost money in their eyes.

Another aspect that seems small, but has definitely an impact on the way of working is the physical environment, according to some managers. Two managers stated that there is often no space to work or sit together (D1, D2). The traditional work environment seems to prevent people from working in a more open and new manner. Another manager is also indicating this as a problem: *“Sometimes an aisle at the floor is already a threshold to approach someone for a question or collaboration”* (A1). The researcher saw also during the visits to the organizations, that at some organizations the total work environment has changed into a **more creative and open place** where people collaborate more frequently. Another manager identified a drawback of this open workspace as due to all the Agile/Scrum meetings, he was sometimes missing silent rooms to work concentrated (A1). Therefore, it is important to pay attention to these **physical factors** at the floor of the organization when implementing DevOps. This can be categorized under the aspect of organizational structures and culture.

When the way of working is changed, the metrics have to be changed as well, in order to solve the interest conflicts between people and departments. Although a lot of people think velocity is and becomes the new metric of performance within an Agile environment, most of the managers already have the insight that this is not the right way to go. The KPI ‘value’ is stated as highly important as people have to add business value with their tasks. This is also acknowledged by a manager and Accenture expert and he states that it is necessary to translate this KPI in **different metrics at different stages** of the implementation. (B1)

Technology

Next to that, it was stated that some organizations **build new technology aside from their old monolithic systems**. This is often done in order to be able to deliver on-going services during the change management and customers notice the least from their change to a new system or technology. The new system is in that case build next to the old one and once the new system is mature enough, they remove the old system. However, related to the application and infrastructure layers, it is often seen that still **some infrastructure parts of the old system keep on existing** due to the absence of a beneficial business case to replace them. *“For some parts of a large existing company, you do not necessarily need Infra as a Code for example”* (G2).

Change management

Large IT service organizations have very often many employees that work already for a longer period at the organization. In order to take them along in changes, **they have to understand the reason to change**. *“Sometimes people work already for more than 20 years in the same way and all of a sudden they have to change this, because they do something ‘wrong’ as they would see it. It is hard to explain that they are not doing something wrong, but another method is working better for the entire organization.”* (H1) and *“You have to let people feel the pain, before they get the urge to change something”* (B1).

4.5.3.2. Additional Findings related to the Implementation

Related to the implementation itself, *“there is not a complete implementation plan for DevOps formulated at the beginning. The plan was written on the way. Higher management decided the high level structure and vision and we as middle management we to a certain extend free to design the new way of working further”* (A1). This is often seen at the organizations as it is not possible to prescribe a final design or end situation. Therefore, the implementation of DevOps can be seen as an **experimental and learning process at lower level**. This is also indicated by the following statement: *“It takes time to discover what the best way to go is for each part of the organization, we will see were each part ends up”* (G1). However, **higher management is often initiating the larger, more radical, changes** that are necessary sometimes in order to change something radically.

The way of changing the organization’s structure is mentioned multiple times. Without concluding which organizational model is the best to choose for DevOps, some conclusions can be drawn on how these organizations have approached such changes. Within one case it is said that the reorganization is done with a big bang approach, where implementing DevOps was done more gradually (A1). Another manager is also stating the organic and experimental growth of new roles within the organization (H1). Someone else mentions that people think that certain layers will become redundant, although this is not true in his eyes: *“Roles will only get another interpretation”* (B1). Therefore, the **approach for changing an organization’s structure can be radical or evolutionary**.

When organizations talk about DevOps, it is important to question in which degree it has to be implemented in the large organization. Large IT organizations have often many divisions and IT departments. It should be realized that **DevOps should only be implemented at the areas where it will likely deliver the most value**. The business case should be beneficial enough to change the entire way of working. Many managers are acknowledging this fact and question themselves whether it has to be implemented at all department (A1, B1, E1, G1, G2, H1). DevOps is a trend nowadays and higher management simply desires to apply a new improvement in their origination, but sometimes there is not a clear vision on where it has to be applied. According to the case interviews, the most suitable area for DevOps can be **based on the type of service and its characteristics**. Service in the form of a mobile application for example is more suitable as this infrastructure is already more standardized and flexible. Other types of services, such as internal services supporting the business process, often have less flexible infrastructure. This makes it less suitable for the implementation of DevOps, although new technologies, such as the cloud, can solve this. Also the degree in which DevOps will be implemented within departments of different types of services differ.

An interesting thing to notice is that almost all organizations from the cases are starting the implementation of DevOps at one department or service line of the organization. The idea for the implementation which is seen at most cases is that it is **starts small with a pilot at a department and it will be spread around** in the rest of the organization after a pilot. Most often this is a **department related to an application based service**, such as a mobile application, due to the flexible nature of such services. When the organization is delivering a IT service aside of its main product, the mobile application component of the organization is often used to start with the DevOps way of working. One manager states that the reason behind this is that this department is most suitable for the adoption of DevOps. This relates to the different types of service on which the implementation is based.

Next to that, the **complex landscape** of large organizations has to be taken into account. One manager is also stating that a full transformation towards a DevOps way of working would be a **very large and complex project**. In his eyes this transformation may even be too complex to undertake for the organization due to all type of legacy (G1). This is also indicated by the fact that implementing DevOps is touching each category within a business process change and each category is important, despite of some differences in focus areas over time. Besides that, large IT service organizations have very often outsourced certain tasks or business lines to other parties. Sometimes, the total operational side of a department is **outsourced, even sometimes abroad**. This makes the implementation of DevOps hard. Large organizations have also many **compliance rules and laws** that have to be taken into account. *“We process financial data of customers. To be allowed to process such data, the development and operations departments have to be separated.”* (F1) Merging the two department is therefore for such organizations not as easy as it may seem at the start. The implementation of DevOps can be hold back due to these constraints. Beside the presence of rules and laws, large existing organizations have many **ongoing contracts** that cannot be terminated easily. These contracts can also hold back an implementation. *“We are between continuous integration and continuous development, but we cannot go over the wall to operations due to all kind of ongoing contracts. The whole organization is struggling with such contracts.”* (I1). Therefore, the complex landscape should be taken into account during the process.

4.5.3.3. Sub conclusions

In conclusion, during the analysis of the empirical data multiple additional findings are identified. These additional findings are interesting for the development of the model as these findings make the model more specific for large IT service organizations. Some additional findings relate to categories in the model, which are identified by the analysis of the empirical data on the categories. Other findings are related to the overall implementation at large IT service organizations, which often resulted from analyzing the extra questions on the limitations and barriers of the organizations. All additional findings are presented in Figure 37. The additional findings were discovered during the analysis of the case interviews. This made it impossible to analyze them with all IT managers. Therefore, the correctness of those findings is evaluated by an evaluation session with experts from Accenture. The results of this section will be shown in section 4.6.

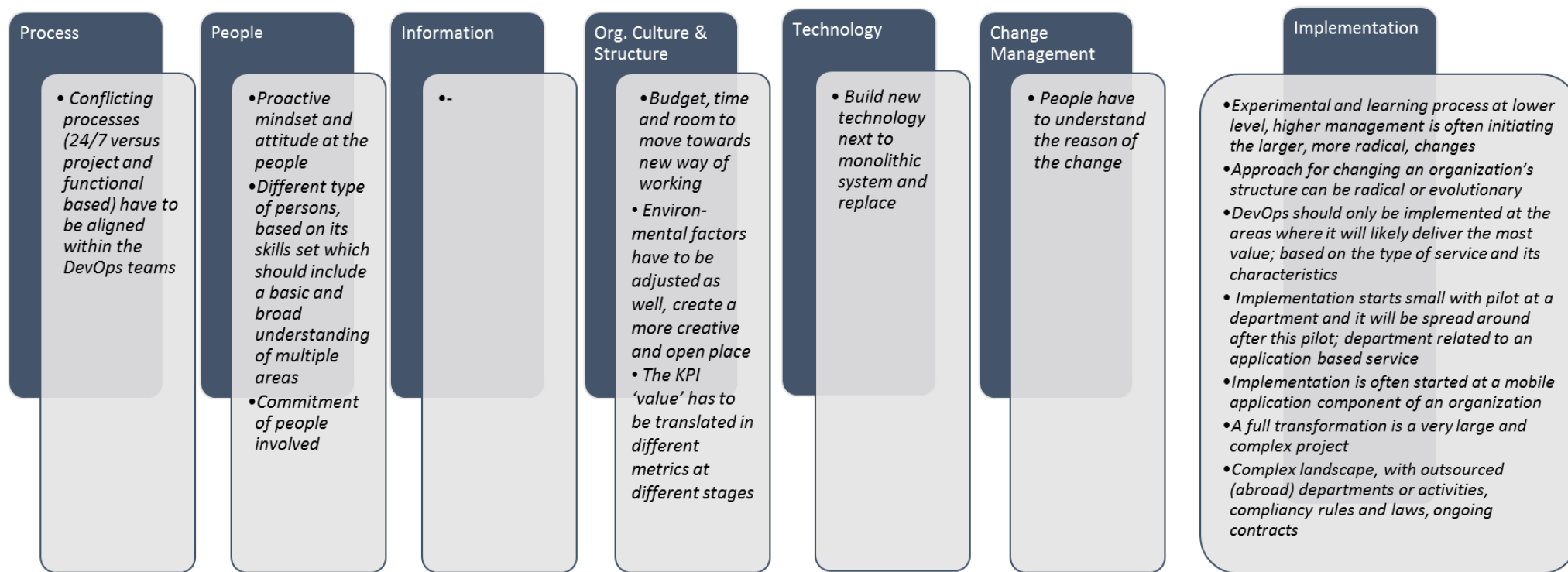


Figure 37: Overview of all additional findings

4.6. Evaluation of the Additional Findings

The additional findings of the previous sections are based on the case interviews. These findings are mentioned or noticed separately of the aspects which are identified in the theoretical model. The additional findings are new within this research and only mentioned by at least one interviewee as an important aspect to take into account for large IT service organizations. However, the additional findings are not questioned within each case interview. In order to evaluate the additional findings, an evaluation session with experts at Accenture is done as described in the research method section 4.2 and the expert selection in section 4.3. Based on the additional findings, statements are formulated and used during an evaluation session.

This section elaborates on the results of the evaluation. First, all statements and answers are shown in a clear overview in section 4.6.1. Also additional explanation is given after the table on the agreements and disagreements on the statements. Section 4.6.2 describes the discussion points that remained after the evaluation session. Section 4.6.3 concludes with the summarized answers of the evaluation.

4.6.1. Introduction of Session

The session was done in combination with a DevOps awareness presentation. In order to introduce DevOps and the research to the audience, some slides were shown and some example questions were asked to get everyone at the same starting point. With these questions, using the tool was also tested to be sure all experts were able to participate.

First of all, the audience was asked to type in words to describe what DevOps is to them. The results are shown in Figure 38. The bigger a word is, the more often it is mentioned by the experts. As the group experts had quite a technical background, it could be expected that they would give answers that are technology related. However, as can be seen in the figure the word ‘culture’ is the biggest, which confirms again the cultural aspect of DevOps. Next to that, ‘speed’ is also named very often. This relates to the objective of DevOps as it promises an increasing time to market with a faster delivery process. The word ‘development’ is mentioned also often, but also the combination of the two departments which indicates the aspect of bridging the two components. All other words are related to DevOps and in line with the research.



Figure 38: Overview of the answers on the first question, visualized in a word cloud

Next to that, the same question was asked to the experts as to the interviewees: “*What would be your main focus area when transforming your organization to DevOps?*”. The experts could divide 100 point over the six categories by using the online tool. The results are shown in Figure 39. Although the focus areas are not completely in the same order as the answers of the interviewees, it shows that at least the ‘winning’ categories are also scored as highest among the experts. Both technology and information are scoring similar scores and are among the lowest. It is again interesting to see that even technology

experts do not see technology and tooling as a main focus point and put the human factors above applying new technical innovations.

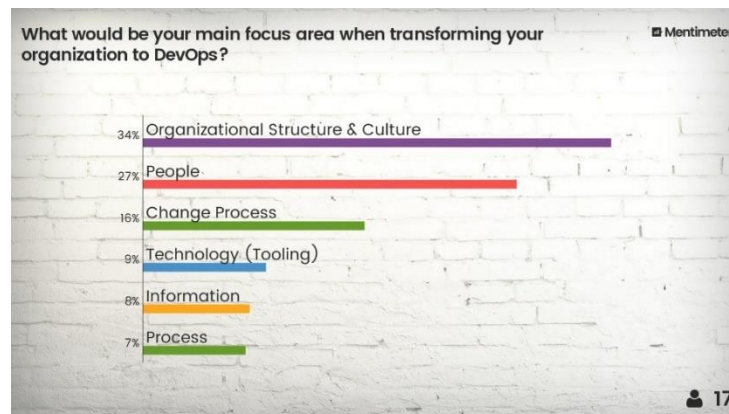


Figure 39: Overview of the answers to the question on the focus areas of DevOps

Next to those two questions, some “example” statements were presented which relate to the approach of the research. This was done to let the experts get familiar with the method and the real statements that would follow and to get the interaction of the audience started. The example statements are stated in Table 6.

Table 6: Example Statements

	Example Statement	True	False
1	You have to work Agile to implement DevOps	50%	50%
2	A special group or team is needed to guide the DevOps transformation	100%	0%
3	If you implement DevOps, there is no way back	100%	0%

These example statements are not used for the actual evaluation, however, it is interesting to interpret the results in more detail.

The first example statement resulted in a balance with exactly 50 / 50%. Some experts were agreeing with this statement, however, others were also stating that DevOps practices could also be applied to the waterfall method or another way of working within development. The research approaches the implementation of DevOps as an extension of the Agile way of working. This is in line with the point of view of Accenture. A DevOps expert explained this from his point of view: “*some DevOps practices are just applied to an organization and you are not working DevOps. Some people call this also DevOps, but that is just dependent on the perspective you have on DevOps, as there is no consensus on the definition and practices.*”. The second example statement was to fully agreed by all expert. As Accenture is a consulting firm and it helps organizations with such transformations, it was not surprising that everyone was stating that a special group or team is needed for this transformation. Next to that, at the last example statement, all experts stated that there is no way back when organizations implement DevOps. However, during the discussion on this statement it was noticed that the experts interpret this statement as: “If you implement DevOps, you do not want to go back”. The reason for this is that they see the benefits of DevOps and they could not think of organizations would want to go back once they work in a DevOps way.

After these first questions and example statements, the presentation was continued. During the presentation, the evaluation statement was presented in between.

4.6.2. Statements

The statements for the evaluation session were formulated based on the additional findings (Table 7). Not all additional findings are evaluated within a separate statement due to time constraints. The statements are mostly based findings related to the implementation and the organizational constraints

that are found. The statements are related to the additional findings as much as possible. However, the objective of the session was to raise a discussion on the topic, the statements are formulated in a more provoking way. The way of formulating varies on purpose from positive to negative. The experts were in this way more triggered to give their vision and comment on the statements. The results are already shown in the table, which will be discussed in section 4.6.3.

Table 7: Overview of all statements and the given answers

Additional findings		Answers		Category
Finding	Statement	True	False	1, 2
- Proactive mindset and attitude at the people; - Different type of persons, based on its skills set which should include a basic and broad understanding of multiple areas	You have to hire different kind of people in a DevOps environment	73%	27%	1
- DevOps should only be implemented at the areas where it will likely deliver the most value: based on the type of service and its characteristics	DevOps (IaaS) have to be applied to all infrastructure of an organization	24%	76%	1
Approach for changing an organization's structure can be radical or evolutionary	Our clients need a complete reorganization to be able to work in a DevOps way of working	33%	67%	1
There are many compliancy rules and laws	Different compliancy rules and laws are limiting organizations from adopting a DevOps way of working	92%	8%	1
Implementation starts small and it will be spread around after this pilot	Organizations have to adopt a FULL DevOps way of working. There is no such thing as half DevOps	9%	91%	1
The implementation is an experimental and learning process	DevOps is not implemented in clear phases, but in a continuous experimental process	91%	9%	1
- Budget, time and room has to be given and commitment has to be there from the top and also the team; - Higher management is often initiating the larger, more radical, changes; - People have to understand the reason to change in order to get them along the change	A DevOps transformation should be approached from bottom-up, in the following order: 1. Individual, 2. Team, 3. Organization	13%	87%	2
Environmental factors have to be adjusted as well, create a more creative and open place	The physical environment has to be adjusted in order to encourage a DevOps way of working	64%	36%	2
Organizations have ongoing contracts	Large organizations have too many existing contracts and legacy to adopt DevOps	7%	93%	2

4.6.3. Discussion of the Outcomes

Two categories are possible to define: 1. Agreements: statements that are confirmed by the experts (4.6.3.1) and 2. Discussion points: Statements in which the discussion did not end with a consensus or comments on the statement were made (4.6.3.2).

4.6.3.1. Agreements

The experts accept with a convincing majority the following statements. First, the experts agreed that different kind of people have to be hired within a DevOps environment. With different people is meant different skills of people that work according to DevOps. However, it was also noted that was ideally the case. It was stated that the actual hiring of completely new people does not happen within large organizations, although DevOps skills can be taken into account in the HR model for hiring new roles if necessary. Therefore, it is indicated as a nice-to-have, where it means to say that it does not have to be implemented, although hiring people with a DevOps mindset can increase the DevOps implementation.

Secondly, it was agreed upon the fact that DevOps should not be applied to all infrastructure within a large organization. It is acknowledged by the experts that a certain level of infrastructure could be held traditionally. One important reason for this was that there is often simply not a business case to transform all current infrastructure into flexible infrastructure. Although it is within the DevOps mindset desired to use new technologies, the infrastructure experts realize that this is not a must do when implementing DevOps in a large organization.

Also, the fact that organizations do not need a complete reorganization for DevOps was confirmed by the experts. With this statement it was mentioned that the word ‘complete’ made the answer easy to make. Most experts agreed that organizations often need some kind of reorganization in the involved departments, but not in the entire organization.

Next, limitations by compliancy rules and laws for organizations do exist within the large organizations. Such aspects are limiting, but it does not mean that those organizations cannot adopt a DevOps way of working. As said before, DevOps has to be implemented in a suitable way for a specific organization in order to take such limitations into account.

To the statement of the implementation of DevOps being a continuous experimental process almost all experts were answering ‘true’. They confirmed that DevOps is not implementable with a straight forward plan and it should be adjusted to the specific organization and its vision.

In line with this, it was also acknowledged that organization do not need to implement a full DevOps way of working, but can also adopt DevOps in different degrees and manners.

Based on these results, the additional findings are assumed to be valuable for large IT service organizations. Therefore, the findings within this category are added to the model that is being developed.

4.6.3.2. Discussion points

On the following statements there was not one convincing consensus among the experts.

Most of the experts seem to disagree that DevOps should be implemented in the order from self to team to organization level. Automatically it could be expected that they mean the other way around, top-down. During the discussion that followed after providing the answers, it seemed to be that the experts do agree with the bottom-up approach. However, they argue that it should be initiated from the top and supported by the management. Without this support and leadership, the DevOps way of working will never be initiated in a successful way. This will also be the only way to move people towards a new way of working. Therefore, you could state that top-down and bottom-up should be combined according to the experts. This is in line with the conceptual model of the research, where those two approaches are indeed combined. A DevOps expert mentions that: *“it should be combination of top-down and bottom-*

up and this should be done simultaneously. You have to take people along the change management but a way to do this is giving them the opportunity from the top.” This confirms the combination of a top-down and bottom-up approach of the model.

Organizations having too many existing contracts and legacy is being denied by the experts. They do not argue that organization's legacy and contracts will completely prevent organizations from implementing DevOps. However, it is stated by the experts that these aspects make the implementation of DevOps more difficult and a challenge to adopt it successfully.

A similar thing applies for adjusting the physical environment. It is agreed by most experts that the physical environment is affecting the success of DevOps. However, there are also some experts that state that a traditional work environment would not prevent a successful DevOps implementation. However, this seemed to change after hearing the story on the creative and open work environment at DevOps departments and that sometimes an aisle or wall is already preventing people to approach each other or another department. The physical wall is often an aspect that is not immediately thought of when talking about DevOps, only the abstract wall between the departments and silos. Based on this discussion, it is decided to perceive this aspect as a nice to have, as it is not mandatory nor irrelevant and it can positively affect the implementation of DevOps.

4.6.4. Sub conclusions

The evaluation session indicated the value of most additional findings. Table 8 and Table 9 present the overviews these evaluated findings. The tables show together the categories that are discussed during the evaluation and indicate how they are evaluated with colors. As can be seen, not all additional findings have been evaluated unfortunately due to research constraints. The additional findings that are not evaluated are however mentioned or noticed by the researcher on frequent basis during the interviews. It can be stated that these findings are acceptable as well as they are often not based on only one single interviewee.

Table 8: Overview of the evaluated aspects by the expert evaluation session

Process	People	Information	Org. Culture & Structure	Technology	Change management
Conflicting processes (24/7 versus project and functional based) have to be aligned within the DevOps teams	Proactive mindset and attitude at the people	-	Budget, time and room to move towards new way of working	Build new technology next to monolithic system and replace	People have to understand the reason of the change
	Different type of persons, based on its skills set which should include a basic and broad understanding of multiple areas		Environmental factors have to be adjusted as well, create a more creative and open place		
	Commitment of people involved		The KPI 'value' has to be translated in different metrics at different stages		

	Relevant for DevOps (Evaluated by Experts)
	Relevant for DevOps (Not evaluated by Experts)
	Nice-to-have

Table 9: Overview of evaluated aspects related to the implementation

Implementation
Experimental and learning process at lower level, higher management is often initiating the larger, more radical, changes
Approach for changing an organization's structure can be radical or evolutionary
DevOps should only be implemented at the areas where it will likely deliver the most value: based on the type of service and its characteristics
Implementation is often started at the application component or at a mobile or internet component within an organization
Implementation starts small and it will be spread around after this pilot
A full transformation is a very large and complex project
Complex landscape, with outsourced (abroad) departments or activities, compliancy rules and laws, ongoing contracts

4.7. Chapter Conclusion

The chapter gives an answer to the third sub question: “*What is the value of the theoretical design for large IT service organizations based on empirical research and expert knowledge?*”. This question indicates the second phase of the Design & Development phase of the research methodology. By conducting case interviews and an evaluation session the value of the theoretical model is being assessed for large IT service organizations and this makes it possible to develop the implementation model into a model specific for large IT service organizations. This value is expressed in multiple sub conclusions that are drawn from the empirical data.

First, the absence of a consensus on the DevOps definition is confirmed by the empirical data. With having a research definition for DevOps, it was possible gather empirical data that was comparable despite the different perspectives of the interviewees. The theoretical starting point of DevOps being mainly a cultural movement is established for large IT service organizations. This results in a common start focus on people and the change management, after that the process and organizational culture & structures and finally, the information and technology. Next to that, due to the competitive environment and the external customer, organizations are driven to implement DevOps. In relation with that, value is the most important measurement to measure the performance of the business.

Moreover, the theoretical aspects of the model are assessed with the empirical data. This resulted in an indication of aspects that relevant for DevOps, aspects that would be nice to have, aspects that already implemented due to an Agile way of working and aspects that are not DevOps specific within large IT service organizations. Additionally, to this indication, a ranking of the relevant aspects to focus on is established by analyzing the results. A detailed overview of these results is shown in Table 11 with its associated legend in Figure 39. Although it is possible to rank these aspects, it should be taken into account that this is not a fixed order that has to be followed for organizations. Therefore, the ranking in the table can be used as general guide to indicate the relevant aspects. The specific order of focus for an organization can vary from this table.

The additional findings during this empirical research relate to the categories defined in the model, but indicate also other important aspects for the implementation approach of DevOps. Most of these findings are evaluated by an expert evaluation session. This resulted in extra aspects that are relevant for DevOps which are added to the model. The findings that are not evaluated by the evaluation session are also added to the model, but with a remark that these are not extra evaluated by experts. The aspects related to the model can be traced back in Table 11 and the aspects related to the implementation can be find in Table 10.

These sub conclusions together contribute to the development of the implementation model specific for large IT service organizations. The empirical data complements the theoretical model with specific insights on the implementation of DevOps at such organizations. These insights apply to large IT service organizations in specific, which can help to develop an implementation model specifically for this type of organizations. This contribution is used within the next section of the Design & Development phase to develop a final implementation model for large IT service organizations.

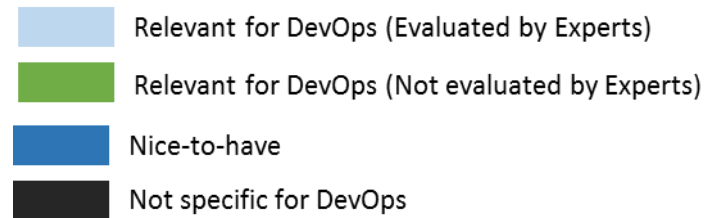


Figure 40: Legend for Table 10 and Table 11

Table 10: Overview of the conclusions upon the implementation

Implementation
Experimental and learning process at lower level, higher management is often initiating the larger, more radical, changes
Approach for changing an organization's structure can be radical or evolutionary
DevOps should only be implemented at the areas where it will likely deliver the most value; based on the type of service and its characteristics
Complex landscape, with outsourced (abroad) departments or activities, compliancy rules and laws, ongoing contracts
Implementation starts small with pilot at a department and it will be spread around after this pilot; department related to an application based service
Implementation is often started at a mobile application component of an organization
A full transformation is a very large and complex project

Table 11: Overview of conclusions upon the categories within model

	Process	People	Information	Org. Culture & Structure	Technology	Change management
Start Focus	2	1	3	2	3	1
Relevant aspects	Continuous Integration	Experimenting without regret	Transparency of org. information	KPIs	Automation	Treat employees equally
	Continuous Delivery	Encourage learning culture	Shared information	Incentives	Virtualization	Feedback sessions
	Systems Thinking	Willingness	Monitoring	Change in organizational structure	Build new technology next to monolithic system and replace	Create understanding
	End-to-end responsibility within self-organizing teams	Cross-skilled people	Standards for coding	Top management support & commitment	Design for Failure	Continuous improvement
	Focus on value	Sharing knowledge		Culture of collaboration	Continuous Deployment	Show results
	ITIL processes in Dev processes	Proactive mindset and attitude at the people		Budget, time and room to move towards new way of working	Focus on MVP	Training
	Balance between dev, innovation and problems	Different type of persons, based on its skills set which should include a basic and broad understanding of multiple areas		Environmental factors have to be adjusted as well, create a more creative and open place		Assign group
	Conflicting processes have to be aligned within the DevOps teams	Commitment of people involved		The KPI 'value' has to be translated in different metrics at different stages		Make an overview of automation
	Sprints	Cross- functional teams				People understand the reason of the change
	Feedback loops					
	Remove non-value adding activities					

Chapter 5

Development of the Final
Implementation Model

5. Development of the Final Implementation Model

5.1. Chapter Introduction

This chapter provides the answer to the sub questions “*How does a model look like for implementing DevOps in IT service organizations and how can this contribute to the implementation of DevOps?*”. The first part of the sub question indicates the third and last section of the Design & Development phase of the research methodology. The conclusion of chapter 4 is the combination of the theoretical model with the practical adjustments for large IT service organizations. Based on comparing this conclusion with the theoretical starting point of the research, conclusions will be drawn for the design of the final model. These conclusions will lead to the developed of the final model in this chapter. The second part of the sub question implies the beginning of the Demonstration phase as this will show how the model can be used.

Section 5.2 explains the choices made for the design of the model and section 5.3 presents the building blocks of the model. Section 5.4 shows a visualization of the final model which Section 5.5 explains the use of the model in more detail with describing use cases. Section 5.6 discusses the assessment tool that is designed based on the research. Finally, section 5.7 concludes the chapter by answering the sub question.

5.2. Design of Final Model

In order to design a final version for the implementation model, the conclusions drawn upon the results of the literature research, the case interviews and expert evaluation session are used.

The conclusions from chapter 3 show that the research takes a business process change approach for the implementation of DevOps. A combination of theoretical models is developed as conceptual design of the model with the important categories that are impacted by a change towards a DevOps way of working. Empirical data shows that all categories are evaluated as relevant in relation to the DevOps implementation at large IT service organizations. Within those categories the aspects of the theoretical model are assessed with empirical data and resulting additional aspects are added. The conclusions from chapter 4 also show that there is a common prioritization of the aspects of DevOps, however, the choice of which aspects are implemented and how this is done fluctuate per type of service, per department and per specific organization. The interviews indicate that it is **not possible to place the aspects in specific phases** as they are varying in importance during the whole process and can be implemented whenever an organization is ready for it and at different levels. Maturities of DevOps do exist, although the components can be cross-spread over the phases, which each its own maturity.

Although the categories within the business process change model will not change, the empirical data shows that the way of what and how DevOps is implemented differs. This is mainly stated due to the additional findings on the implementation of DevOps. The starting point of the research was that the implementation of a DevOps way of working have to be approached from a radical and evolutionary perspective. This was one of the reasons to choose for a business process change approach as described in Chapter 3. The interviews show that this is a valid approach, although this is not done according to a fixed plan. The empirical data also indicates the addition of multiple aspects to the model related to the implementation. It is noticed that the transformation towards a DevOps Way of working is not a straight forward implementation process. The revolutionary models assume a radical way of implementing a change in the business process and the evolutionary methodologies approach change as continuous improvement. Regarding the IT service organizations, too many radical changes will interrupt their IT service process too much, where ongoing continuous improvements will never lead to the desired disruptions in the way of working that DevOps desires (Weerakkody et al., 2011). Combining the two approaches leads to a complimentary approach, although it does not see the entire process as a dynamic one. In contrast, the implementation of DevOps is very dynamic and complex. As a DevOps way of working encourages a learning culture, this mindset can also be translated to the implementation by stating the **implementation approach has to be adaptive and dynamic** in itself. Therefore, it seems not logical to implement the DevOps aspects in a fixed and sequential order. Even the IT service performance metrics that are used within different DevOps maturities are dynamic. The focus should be

on adding business value, however, this can be translated in different ways across the maturity of DevOps.

To summarize, the aspects and the implementation approach of the DevOps way of working are both dynamic. All components can fluctuate in level during the implementation. This complexity is not described within the current literature models for business process change. Therefore, in the following sections the building blocks are presented for a DevOps Implementation model for IT service organizations that is based on elements of BPC models together with the empirical evidence from the interviews. These building blocks are: external drivers, components with its aspects, the implementation approach and critical organizational factors. They will be presented one by one and in conclusion it is showed how they can be combined into a DevOps implementation model for large IT service organizations.

5.3. Building blocks

This section presents the building blocks of the model and elaborates on which choices are made for each building block. The building blocks contribute to the research objective as they can be used to help IT managers to focus on the relevant aspects of DevOps. These aspects relate to the challenges which are formulated as objectives for developing the DevOps implementation model. The building blocks can be combined for the development of a complete implementation model for large IT service organizations. First, the external drivers for DevOps within a large IT service organization are explained in section 5.3.1. The core components with its aspects are discussed in section 5.3.2. Section 5.3.3. describes the implementation process aspects and section 5.3.4 describes the critical organizational factors.

5.3.1. External Drivers

As first building block, the drivers for implementation DevOps are identified. The interviews show the main drivers for large IT service organizations to implement DevOps are external customers and industry competitive. Next to that, it is always possible that there are other external drivers, such as laws, suppliers and natural factors, which let organizations decide to implement DevOps, but these occur less according to this research. Within the implementation of DevOps, it has to be acknowledged that technology should not be leading. Technological innovations could drive a transformation towards DevOps, although the implementation should not only be focused on this part. The interviewees indicate that the human factors are the most important to focus on, especially to start a DevOps transformation.

External Drivers
Competitors
Customers
Others

Table 12: Overview of external factors

5.3.2. Components & Aspects

The components of the model are people, process, organizational culture and structures, technology and information and its overall change management, see Figure 41. Based on the literature and empirical data, a DevOps way of working has an impact on each of these six categories. All components are relevant to take into account with a DevOps implementation at large IT service organizations. The component people has a priority as focus area in the beginning, together with the change management component, although all components can be important. The interviews show that the implementation of DevOps will not reach its successfulness when people are not taken into account in the beginning. The components information and technology are facilitators while those components can also be crucial at during implementation.

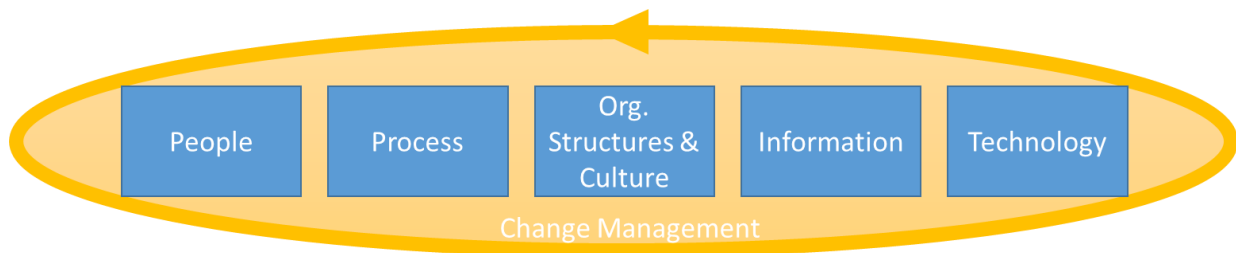


Figure 41: Overview of the components

Each component consists of several key DevOps aspects based on the research. The order in which the aspects are presented is not leading and fixed. Based on the empirical data, it is known that each organization can have its own ranking and choice on which aspect they desire to focus on. However, this ranking is formulated based on the empirical data of the nine cases and could be seen as a common order of aspects. The ranking is based on the current degree and importance for each case and therefore it can be assumed that this is on average a general situation for a large IT organization that is working Agile and slowly moving towards DevOps.

Although each aspect is organization dependent, the aspects presented in grey in Table 13 are highly dependent on current way of working. As an organization works according to an Agile methodology, it is assumed that those aspects are already in place. If this is not the case, organizations have to focus on applying Agile first, before implementing DevOps according to this model. The aspect presented in dark blue in Table 13 are nice-to-have's. This means that a IT manager can decide itself whether this aspect is desirable for his department, where the other aspects are recommended to take into account when possible.

Table 13: Overview of the components and its aspects

	Process	People	Information	Org. Culture & Structure	Technology	Change management
Start Focus	2	1	3	2	3	1
Relevant aspects	Continuous Integration	Experimenting without regret	Transparency of org. information	KPIs	Automation	Treat employees equally
	Continuous Delivery	Encourage learning culture	Shared information	Incentives	Virtualization	Feedback sessions
	Systems Thinking	Willingness	Monitoring	Change in organizational structure	Build new technology next to monolithic system and replace	Create understanding
	End-to-end responsibility within self-organizing teams	Cross-skilled people	Standards for coding	Top management support & commitment	Design for Failure	Continuous improvement
	Focus on value	Sharing knowledge		Culture of collaboration	Continuous Deployment	Show results
	ITIL processes in Dev processes	Proactive mindset and attitude at the people		Budget, time and room to move towards new way of working	Focus on MVP	Training
	Balance between dev, innovation and problems	Different type of persons, based on its skills set which should include a basic and broad understanding of multiple areas		Environmental factors have to be adjusted as well, create a more creative and open place		Assign group
	Conflicting processes have to be aligned within the DevOps teams	Commitment of people involved		The KPI 'value' has to be translated in different metrics at different stages		Make an overview of automation

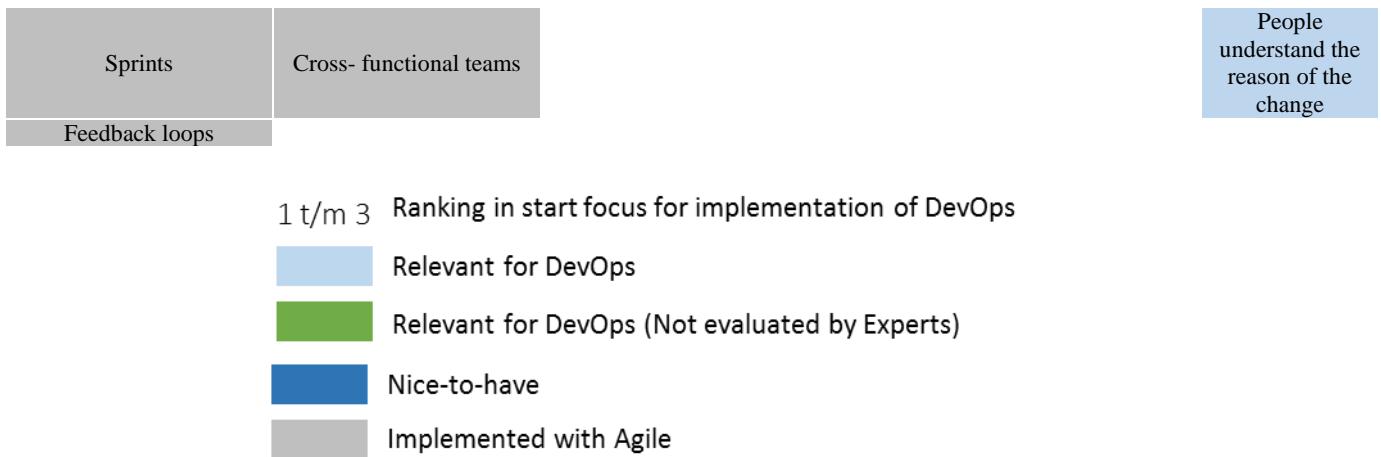


Figure 42: Legend of Table 13

As said, each component has their own maturity. Based on the implementation degree of the aspects, each category has a maturity. Within this building block, this is visualized by a “S”-shaped curve for each category (Figure 43). The curve presents the level of maturity per component. Although the components are interconnected with each other, different maturities are possible. It has to be acknowledged that this figure is an example and the curves can look differently. Nevertheless, it can be assumed that the form of the curve is “S”-shaped, as this relates to the common curve to indicate a learning or adoption process. First, the curve is likely to increase slowly as the changes are not immediately adopted and have to be learned. Eventually, the maturity will increase faster as the adoption goes better and at a certain point the maximum maturity of that component is reached. Associated to the DevOps maturities are the **varying metrics** that will be used for defining the performance. The focus should be on adding business value, however, this can be translated in different KPI across the maturity of DevOps.

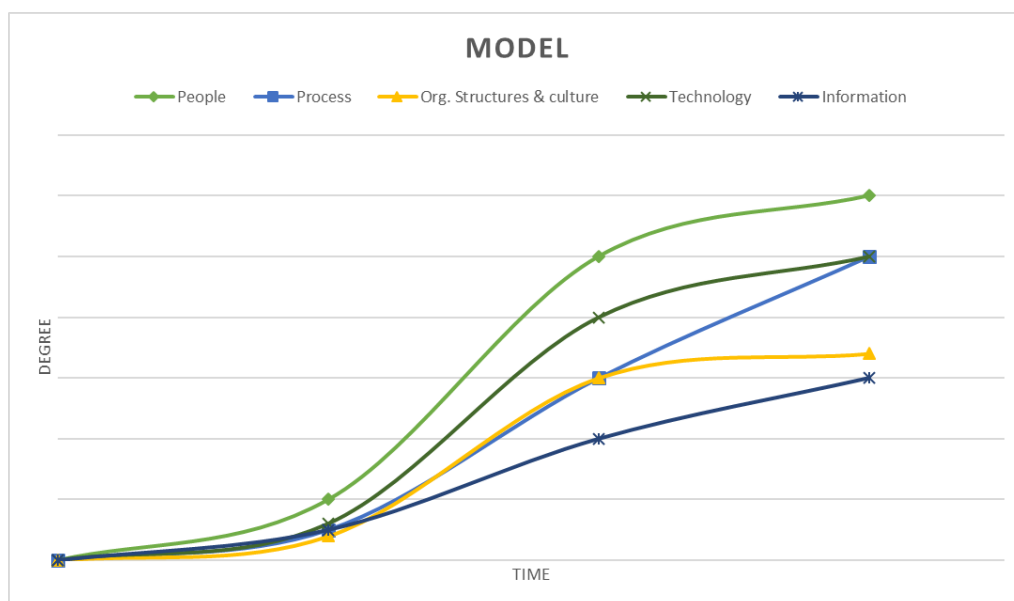


Figure 43: Example of different implementation maturities per component

5.3.3. Implementation Approach

In addition, the interviews yield interesting results on the implementation approach. Based on the aspects that are found for the implementation approach (Table 14), the building block is developed step by step.

Table 14: Overview of conclusions on implementation approach

Implementation
Experimental and learning process at lower level, higher management is often initiating the larger, more radical, changes
Approach for changing an organization's structure can be radical or evolutionary
DevOps should only be implemented at the areas where it will likely deliver the most value; based on the type of service and its characteristics
Complex landscape, with outsourced (abroad) departments or activities, compliancy rules and laws, ongoing contracts
Implementation starts small with pilot at a department and it will be spread around after this pilot; department related to an application based service
Implementation is often started at a mobile application component of an organization
A full transformation is a very large and complex project

First of all, within the organizations people appear to be aware of the varying nature of different types of departments. On the one hand the front office desires to be flexible and adaptive to the changing requirements and on the other hand the back office is trying to be as stable as possible in order to provide a stable service continuously. This results in **different types of services in which DevOps is being implemented**. An application based service is much more applicable for a DevOps way of working than a service that is more infrastructure based. Therefore, many organizations have started their 'transformation' **within a more application based department of the organization**, often the mobile or customer service related department. This start of implementing DevOps can be seen as a **pilot** to prove the concept within the organization. After a successful pilot, the way of working is applied to the rest of the organization. Though, large IT service organizations have many legacy systems within their infrastructure and often do not have a beneficial business case for replacing all infrastructure with new, adjustable infrastructure such as IaaS or applying the Cloud. This should only be done when it is beneficial based on the type of service and organizational characteristics. There will often remain a part of IT infrastructure that will work independent of the DevOps teams.

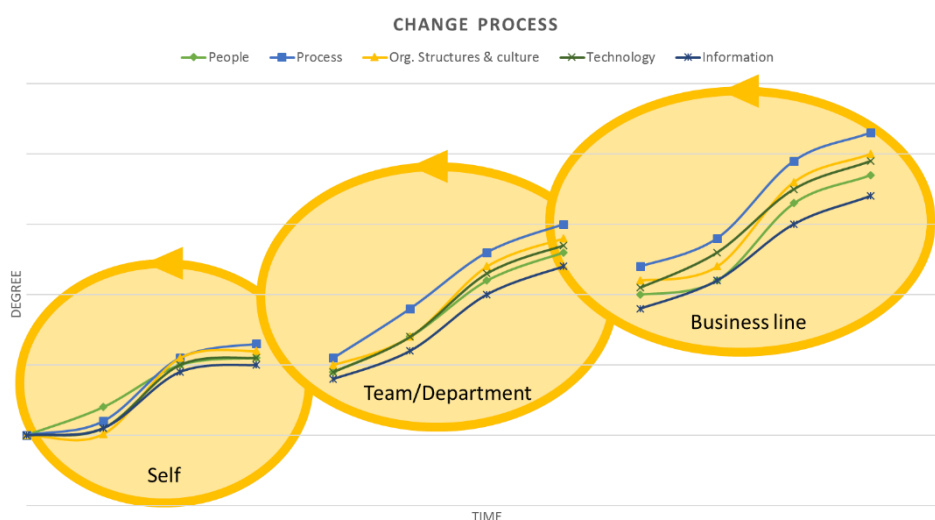


Figure 44: Building block of implementation approach

The approach of implementation will be radical and evolutionary. Top management support and commitment is always important from the start. Changing the way of working will be done continuously, following the S-curve. However, at a certain moment the maximum maturity of the components is reached. This is the maximum of the selected **span of control** of the IT manager. Based on the idea of the pilot and its expanding area, the span of controls “self – team – business line” are defined, where team can also relate to a department. These span of controls can be added to implementation model of Figure 43 to indicate the different level on which the DevOps relevant aspects can be implemented. This is visualized in Figure 44. When the maximum maturity of DevOps is reached, a radical change is necessary to go to the next change. In that case, the “S”-shaped curve will start over at another level in that span of control. This will increase the maturity of DevOps within the organization. However, within the different span of control levels the implementation of DevOps can already be started by continuous improvement at certain aspects.

As the additional findings related to the implementation state, implementation approach has to be focused on experimenting at lower level. There is not a specific plan formulated for the process that needs to be undertaken, but it is important that people can **experiment without regret** during the change management and discover the best way to go. **Learning by doing** should also be embraced within the process as people will only learn on-the-job how to handle the DevOps way of working. With this mindset it is important to keep the focus on continuous improvement as the experimental aspect is not meant to work without any rules or constraints. Therefore, failure should not be judged, but understood and lead searching for a solution.

Large organizations have many laws, contracts, legacy and compliance rules to comply with. Therefore, it is harder to enable such an experimental environment. By **creating exits during the implementation** it will be easier to find out what is the best way of working for an organization. Exits are here defined as moments at which the implementation of DevOps will be stopped as it does not deliver the desired results or the maximal degree or most suitable DevOps way of working is already implemented. At such a moment it can respectively be decided to reverse the process back into the previous structure or keep the current DevOps way of working and not extend this anymore. This also relates to the fact that **DevOps does not have to touch all business areas completely** as mentioned as well for the infrastructure. For example, key business departments and first level operational departments such as marketing and the operational helpdesk will definitely notice the DevOps implementation, but will not directly touched by it.

The empirical research shows that the process of implementing DevOps is dynamic. During the entire process, it is examined and researched which way is followed. It is not possible to appoint a specific standardized path that every organization can follow for the DevOps implementation. It is highly dependent on the organizational context and factors. This is in line with the contingency theory. The contingency theory states that organizational characteristics have to fit their contingencies (environment, organizational size and strategy) in order to lead to high performance (Donaldson, 2006). Therefore, the DevOps implementation has to fit the organizational factors. This is an important aspect to take into account as the degree on which DevOps is implemented in an organization is affected by these factors. Therefore, a separate building block is developed for the organizational factors, which is described in section 5.3.4.

5.3.4. Organizational Factors

The interviews show critical organizational factors that have direct influence on DevOps possibilities. Referring to the contingency theory again, the organizational context and factors are relevant to take into account for the DevOps implementation. Therefore, during the implementation process these factors have to be taken into account continuously as this was also acknowledged by the experts. As can be noticed, the overall organizational culture is adopted in these factors. The organizational structures and culture component is different from this as there is always an overall culture present in an organization, that is hard to change.

Table 15: Overview of organizational factors

Organizational factors
Physical environment
Complex landscape due to technology, stakeholders, legacy
Ongoing contracts
Many compliancy rules and laws
Different tasks are outsourced, sometimes even abroad

5.4. Presentation of Final Implementation Model

The building blocks gather all relevant factors for the implementation of DevOps in a large IT service organization. The building blocks identify the most relevant focus areas and aspects, but do also identify important characteristics for the implementation approach. By composing the building blocks together, a final implementation model for DevOps in large IT service organizations can be designed. Table 16 presents the final overview of all relevant aspects in relation to the DevOps implementation in a large IT service organization. Within Table 16 also the aspects are indicated that would be nice to have and aspects that are expected to be already implemented due to an Agile way of working. Figure 46 presents a simplified visualization of the model for the implementation approach that is related to the relevant aspects. The building blocks show that the implementation of DevOps is a complex and dynamic process. Due to this complexity, it is also hard to visualize all building blocks into one model. Therefore, a couple of important factors that are not displayed in the model have to be mentioned and these comments are associated to Table 16 and Figure 46. These three components present the final implementation model for the implementation of DevOps at large IT service organizations that work already according to an Agile way of working.

Associated comments of implementation model of Table 16 and Figure 46

- **Working Agile**
The model assumes organizations work already according to an Agile methodology. If this is not the case, the grey aspects in Table 16 can indicate which Agile practices are important to get in place first, as well as other possible Agile practices, before implementing DevOps.
- **Start focus**
The focus areas for the start of the implementation is not visible in the figure, only in Table 16. The common start focus is on the components people and the change management, after that the process and organizational culture & structures and finally, the information and technology.
- **Aspects**
The overview of relevant aspects is given in Table 16. The ranking of the relevant comments can be seen as a common ranking, general to large IT service organizations. Depending on the organizational factors, this ranking has to be made organization specific by the IT manager. Next to that, the aspects indicated as nice to have can be seen as aspects that large IT service organizations will not focus on in the first place, but implement it when it is possible and a high degree of the other aspects is already in place.
- **Component curves**
Each component has its own curve. The curves are formed on basis of the degree of implemented aspects of Table 16. Per component the list of aspects can be filled in and the total degree can be determined, which will affect the height of the curve per span of control. The way on which the curves are formed present the adoption process of DevOps aspects in each component. As the most important performance measurement is value, the focus should always be on adding business value. This has to be translated in different KPIs across the maturity of DevOps.
- **Radical & continuous approaches with exits**
Between the different spans of control, radical changes are necessary in order to gain more mature in DevOps according to the model. However, it should be noticed that before a radical change is done towards a next span of control, the components at other levels can already be improved on continuous basis. This is visualized by the overlapping circles. Next to this, there are during the whole process exits in which the implementation can be stopped. Due to the experimental and learning by doing process in which DevOps has to be implemented, it always has to be possible to stop. Otherwise, the blameful characteristic will limit this experimental process approach.
- **Pilot & type of services**
For the DevOps implementation at large IT service organizations a pilot is often started at an application based service of an organization. This is not visualized, although it is an important factor as such services suit best for DevOps. By starting at this part of the organization, the lessons learned can be used during the implementation in the rest of the teams and business lines. The areas in which DevOps will be implemented is depending on the type of service that is being delivered by the business lines. Often a certain degree of the infrastructure, operations and business departments are not integrated in the full DevOps way of working due to the absence of a beneficial business case for the transformation towards DevOps at those areas.

Table 16: The relevant focus areas and aspects of DevOps implementation

Process		People	Information	Org. Culture & Structure	Technology	Change management
Start Focus	2	1	3	2	3	1
Relevant aspects	Continuous Integration	Experimenting without regret	Transparency of org. information	KPIs	Automation	Treat employees equally
	Continuous Delivery	Encourage learning culture	Shared information	Incentives	Virtualization	Feedback sessions
	Systems Thinking	Willingness	Monitoring	Change in organizational structure	Build new technology next to monolithic system and replace	Create understanding
	End-to-end responsibility within self-organizing teams	Cross-skilled people	Standards for coding	Top management support & commitment	Design for Failure	Continuous improvement
	Focus on value	Sharing knowledge		Culture of collaboration	Continuous Deployment	Show results
	ITIL processes in Dev processes	Proactive mindset and attitude at the people		Budget, time and room to move towards new way of working	Focus on MVP	Training
	Balance between dev, innovation and problems	Different type of persons, based on its skills set which should include a basic and broad understanding of multiple areas		Environmental factors have to be adjusted as well, create a more creative and open place		Assign group
	Conflicting processes have to be aligned within the DevOps teams	Commitment of people involved		The KPI ‘value’ has to be translated in different metrics at different stages		Make an overview of automation
	Sprints	Cross- functional teams				People understand the reason of the change
	Feedback loops					

1 t/m 3 Ranking in start focus for implementation of DevOps

- Relevant for DevOps
- Relevant for DevOps (Not evaluated by Experts)
- Nice-to-have
- Implemented with Agile

Figure 45: Legend associated with Table 16

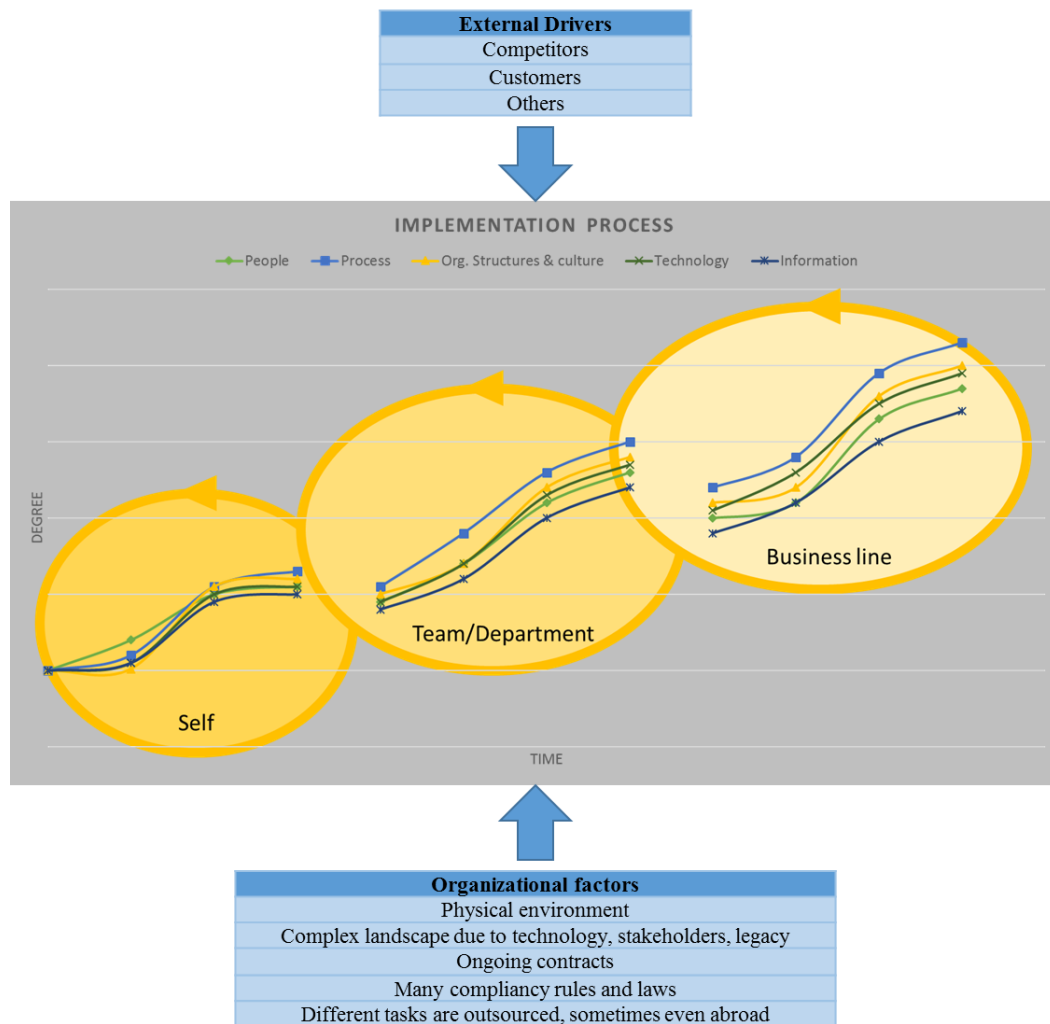


Figure 46: Simplified visualization of implementation model

5.5. Assessment Tool

The interview guide that is used during the case interviews seems to be a helpful tool for organizations that are thinking of adopting a DevOps way of working. Therefore, the interview guide is used to develop an assessment tool, which can be found in Appendix H – Implementation Assessment Tool. Moreover, such assessment is often recommended to be done, at least before a radical change. During the continuous learning phase an assessment is not necessary, however, for a radical change it has to be clear what the current situation is. The guide can create a clear overview of which aspects an organization has currently implemented. In order to do this, the aspects that are mentioned within the interview guide are updated according to the results of the research. The updated interview guide can be seen as an assessment tool for organizations to discover which aspects they have already implemented. Also the aspects on which they would like to focus the most can be indicated by ranking its importance. In this way the current situation and vision on the DevOps within an organization can be formulated. Therefore, this guide can help Accenture to get more insight into the wishes and requirements of their clients. The tool can also be used within one organization to get insight on how the involved people are perceiving the situation and the way to go when talking about DevOps. This can be done by letting everybody fill in the form of the tool and compare it with each other. It may occur that multiple people within an organization have a completely different perspective on what is happening within the organization. This tool could start an open discussion between the involved people on where the organization currently is standing.

5.6. Demonstration

With presenting the final implementation model and the assessment tool within Section 5.4 and 5.5, the Design & Development phase of the research methodology is finished. Following the research flow diagram, the Demonstration phase is started as next step. The second part of the sub question “*How does a model look like for implementing DevOps in IT service organizations and how can this contribute to the implementation of DevOps?*” focuses on the use of the developed implementation model. In order to provide an answer to this question, multiple use cases are used as illustrations on how the model can contribute to the implementation of DevOps at large IT service organizations. The implementation model states that the implementation of DevOps is situational to multiple organizational factors. The organizational factors defined in the model are used to formulate different use cases, which are inspired by the cases that are used within this research. Sections 5.6.1 to 5.6.4 will illustrate each a use case related to these organizational factors, where section 5.6.5 illustrates the use of the assessment tool.

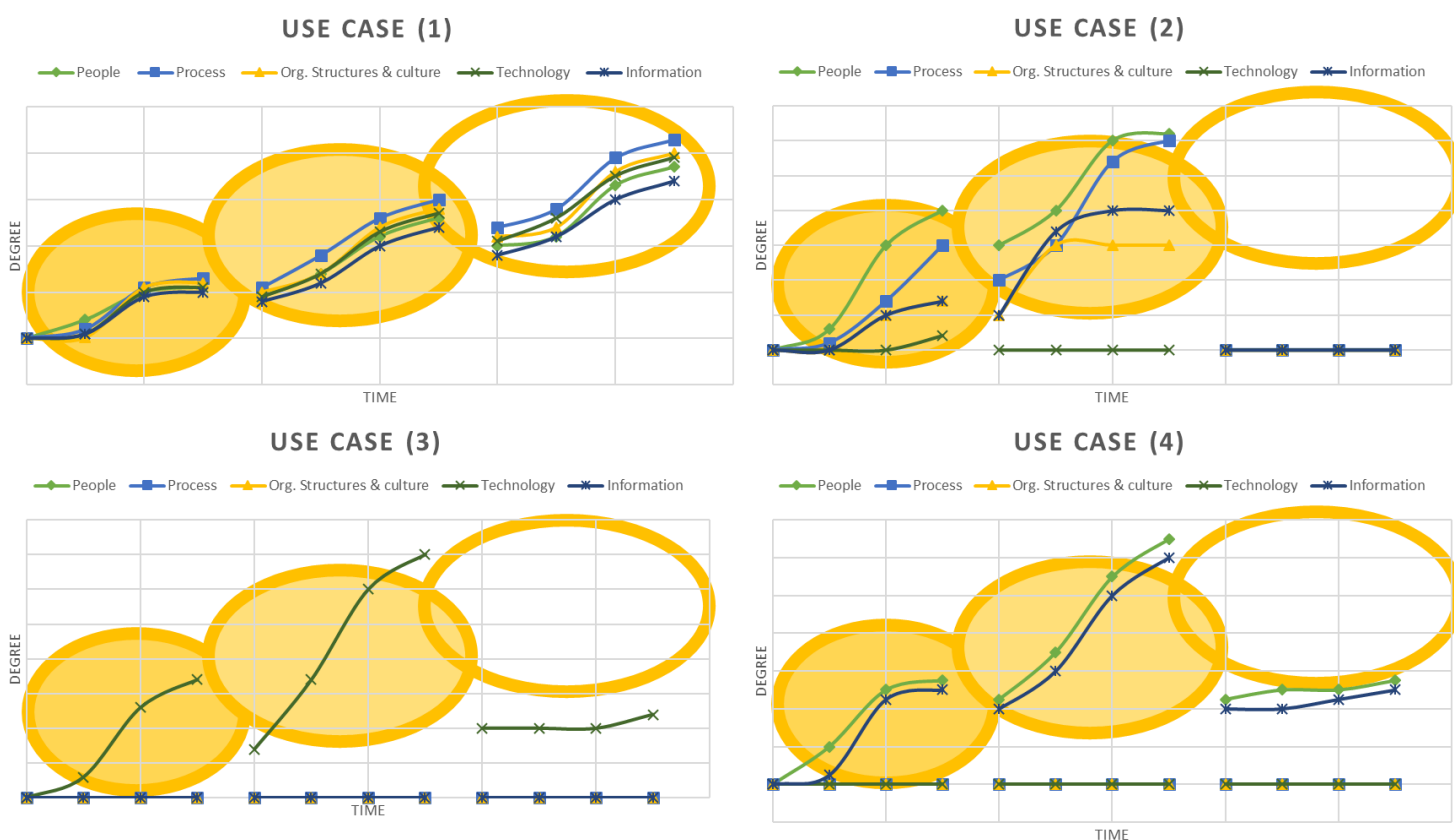


Figure 47: Overview of the visualized model for the use cases

5.6.1. Use case 1 – Complex Landscape & Legacy

Situation: An organization wants to make the delivery process of their mobile application faster. In this case problems with the application experienced by the user can be solved at a higher speed. A IT manager states that DevOps have to be implemented, however, he is bounded due to the complex landscape and the high amount of legacy of the organization. Therefore, it is decided by higher management to decouple this department of the rest of the organization, which makes it less dependent on the organizational factors and creates a whole separate business line for the mobile application.

Model: The model in Figure 47 for this use case shows clear S-curves for all categories within each span of control. The department is now a separate business line which creates the opportunity to adopt a DevOps way of working from scratch. DevOps aspect can be implemented first within **single tasks or persons**, where after the **team** will be working in a DevOps way of working and eventually the whole

business line is integrated in this DevOps way of working. The curves follow a common adoption line which is not bounded by organizational factors, but presents the normal **learning characteristic** per category. This can be indicated as the ideal situation of the developed implementation model. However, this is not likely to happen often, as the organizational structures do affect the process and organization can often not choose to decouple a whole business line. This will be shown in the other used cases.

5.6.2. Use case 2 – Current Culture & Physical Environment

Situation: Suppose a large IT service organization where an IT manager is responsible for a team which maintains a medium maturity of an Agile way of working. However, the organization has still a traditional organizational culture among its employees and it is not likely that the physical environment can be changed. The IT manager wonders how to implement a DevOps way of working despite this traditional culture of his organization.

Model: The implementation model can help the IT manager to indicate on which aspect he should first **increase the Agile way of working** in order to implement DevOps within his team. After that, as the mindset of the employees is still traditional, it is important to **focus first on the people** component, which the manager can derivate from the start focus in implementation model. The implementation curve in Figure 47 shows the degree of implementation per category of the DevOps implementation. The focus seems to be placed on the people component, as this has the largest learning curve. Together with this, the **process** component has also been impacted at this level. This could be explained with the higher degree at the people component as they can influence the process in a positive way. This is continued at the department level, where also the other components are more affected by the implementation of DevOps. The next step, the implementation of DevOps regarding the whole business line will not be made as the **overall organizational culture and structures are limiting** this. In this way, the implementation of DevOps is based on the lower level.

5.6.3. Use Case 3 – Compliancy

Situation: Suppose an organization operating in the financial industry. The IT manager knows the benefits of DevOps and desires to implement it within his department. However, his department processes financial data of their customer and has to cope with compliancy rules related to the customer's privacy. A compliancy rule that exist for organizations that progress financial data is that people dealing with operational activities cannot be involved with development activities. This can limit the implementation of DevOps, however, the implementation model shows that this does not prevent the organization from implementing DevOps.

Model: As a DevOps way of working can be adjusted according to the organizational factors, the implementation model can indicate on which other aspects the DevOps implementation can be focused. The compliancy rules limit the **organizational structure** and **processes** components to change, however, the technology component can create opportunities for this. New technologies, such as automation, may provide opportunities to work together without merging the teams together and information aspects could improve the communication between the departments. The implementation curve is shown for the **technology** component in Figure 47. First the technology aspects are increased at the single tasks, then the implementation of a new innovative technology that is compliant to the rules is implemented at team level. After that, the technology aspects could be adopted at **business line level**.

5.6.4. Use Case 4 – Outsourcing

Situation: Another organization is operating in the retail industry and does not have a large IT department. Therefore, the organization has outsourced most of its operational activities to other parties. This limits the possibilities of implementing DevOps related to the process and technology components, as the organization cannot decide on the implementation of DevOps for the operational activities. Also the organizational culture and structure cannot be changed as there are multiple stakeholders involved. However, the organization itself can look for opportunities within the components information and people to align their development with the operational activities at their partners.

Model: The implementation curves in Figure 47 show that the organization is focused on the **information** and **people** components. This can help to align the **process** as for example the degree in which the information is being shared between all stakeholders in the business line is improved. Focusing on the people and their mindset can improve the collaboration in the business line as well, although this will also be influenced by the stakeholders. Also specifications in the contracts related to DevOps could improve the adoption of a more DevOps way of working despite the outsourced operational department. Eventually, the model may trigger the IT manager, in agreement with higher management, to choose for **less outsourced tasks** in order to implement DevOps in a higher degree, when this is expected to be beneficial. This could be seen as **more radical change** that could increase the degree of DevOps at business line level.

5.6.5. Use Case 5 – Assessment

Suppose a IT manager at a large IT service organization with different types of services and that wonders **where to start with its DevOps implementation**. The type of service can influence the way on which DevOps has to be implemented, such as services that support the business process itself and services that are provided to the external customer. The implementation approach can **differ within one organization itself as it has to fit the type of service**. The model can help to indicate that a pilot within a mobile application department is more suitable to start. In this case, the **assessment tool** is very useful. The IT manager can use this tool to **identify the current way of working regarding DevOps**. He can also ask the people involved to fill in the tool to identify the **different perspectives** on the current way of working within the specific departments.

5.7. Chapter Conclusion

This chapter provides the answer to the fourth sub question: *“How does a model look like for implementing DevOps in IT service organizations and how can this contribute to the implementation of DevOps?”*. The first part of this sub question relates to the last section of the Design & Development phase of the research methodology and the second part relates to the Demonstration phase that is followed. The last section of the development contributes to the research by creating a final model as for the implementation of DevOps at large IT service organizations based on the conclusions of the theoretical and empirical research. The demonstration phase is used to illustrate the use of the final model.

Based on the conclusions of chapter 4, the aspects and the implementation of DevOps are dynamic. The categories that are defined based on the literature are defined as relevant, although the focus on these categories can vary. It is also not possible to place the aspects in specific phases as they are varying in importance during the whole process and can be implemented whenever an organization is ready for it and at different levels. Next to that, the implementation is also dynamic and cannot follow a straight-forward plan as the degree in which DevOps will be implemented is highly depending on organizational constraints. Due to this, it is concluded that the dynamic characteristics of both the relevant aspects and their implementation approach is not described within the current literature models for business process change.

Therefore, building blocks are formed on basis of the conclusions of chapter 3 and 4. This resulted in four building blocks which are 1) External drivers, 2) Components and its aspects, 3) Implementation approach and 4) Organizational factors. The building block external drivers describe the competitors and customers as the main reason to implement DevOps. The building blocks components and its aspects describe relevant focus areas for the DevOps implementation and states that the components people and change management are the most important to focus on in the beginning, where after process and organizational culture and structures have to be taken into account and as last the information and technology. Next to that, it provides an overview of the aspects that can be categorized in relevant, already implemented due to an Agile way of working and nice-to-have's. Also the degree of implementation per component is determined by these aspects and can be shown with component curves, which likely look similar to the common adoption curve. The building block implementation includes all aspects that are determined regarding the implementation approach, which is partly visualized by

different spans of control. The other aspects are too complex to visualize. The fourth building block organizational factors is a very relevant one as this includes the organizational characteristics that can affect the implementation of DevOps, in both its aspects and implementation approach.

The building blocks are combined in a final model which is partly presented by a visualization (Figure 46). The relevant focus areas and aspects are shown in a clear overview in Table 16. As the whole implementation approach is very complex and dynamic, depending on multiple organizational factors, the model is a simplified visualized and additional factors are given in text together with the model. These additional comments relate to the current way of working that has to be Agile, start focus, aspects, component curves, radical & continuous approaches with exits, pilot & type of services. Next to that, an assessment tool is developed based on the interview research model which can be used associated to the model to assess the current state of the relevant aspects as is also explained in a use case. The use of the model is explained by illustration four use cases, each describing another organizational situation, regarding complex landscape and legacy, current culture and physical environment, compliancy and outsourcing. This indicates that the implementation of DevOps is highly dependent on the organizational factors and the implementation model that is developed can help to focus on the relevant aspects. The implementation model also states relevant aspects for the implementation approach.

Chapter 6

Conclusion

6. Conclusion

6.1. Chapter Introduction

This chapter finalizes the research report with the most important conclusions and discussions. First, the conclusions are discussed by first answering the sub questions one by one, which results in an answer to the main research question (section 6.2). Also the scientific and societal contributions are being discussed. Section 6.3 discusses the limitations of the research and the used research methods. Section 6.4 reflects upon the research discusses the choices made during the research and its relation with the master program for which the research is conducted. The limitations and reflections are followed by addressing future research possibilities with research questions in section 6.5. Finally, recommendations are done towards Accenture on the use of the developed model.

6.2. Conclusions

Within this master thesis project a research is performed on the implementation of DevOps in large IT service organizations. The main research question is:

How does a guiding model look like to support the implementation of DevOps in IT Service organizations?

In order to answer this main question, a design science research is performed in which four sub questions have been formulated. The answers to those sub questions are first discussed before answering the main question.

Sub question 1: What is meant by a DevOps way of working and how is this related to currently used methodologies within IT service organizations?

The answer to this question was retrieved by doing extensive literature research on DevOps and related concepts and by conducting explorative expert interviews at Accenture. First, IT service organizations were examined by looking at their service delivery process and associated methodologies. This resulted in an elaborative domain description of related methodologies before and included in DevOps. Based on this domain and more literature research, a research definition of DevOps is formulated.

The definition of DevOps used within this research is:

“DevOps is a cultural movement that breaks silos between the business, development and operations department, combined with a number of service development practices regarding people, process and technology, that enable rapid development and delivery of services”

Additionally, DevOps is seen within the research as an extension of the Agile methodology. It can be seen as a response to the inadequate collaboration between the two departments development and operations. DevOps combines several practices of Agile, Lean and ITSM and applies it to the whole delivery pipeline. Using the research definition as starting point, the values and practices are identified. The values of DevOps are defined as systems thinking, focus on flow, amplify feedback loops and culture of continual experimentation and learning, in relation to the model of Kim (2013). Also the key areas in which challenges of DevOps occur are identified as people, process and technology. These areas are seen the objectives of the solution and have to be involved in the model. The areas people, process and technology relate to the common key elements of process improvement, according to Prodan et al. (2015).

Sub question 2: What type of change model for the implementation of DevOps should be used and how does a first design of the implementation model for a DevOps implementation look like?

The theoretical design of the model is developed with answering this second sub question. The research approaches the implementation to a DevOps way of working as a business process change due to the relation of the objectives of the solution related to the process improvement elements. DevOps impacts the whole business process by breaking through silos and focusing on systems thinking throughout the whole process. Literature research has been done to gather theoretical insights on different business

process change models. Regarding the implementation approach, the Business Process Change field include two different types of approaches, the revolutionary (top-down) and evolutionary (bottom-up) perspective. The Business Process Reengineering methodology describes a radical change that is strategic-driven that is focusing on the customer by looking at the relations of the main components of a business process change. Total Quality Management is also focusing on delivering quality to the customer and improves continuously. Lean Six Sigma acknowledges the importance of removing waste from the process by continuous data-driven decisions. Based on the literature and the DevOps definition and challenges, a combination of the most relevant business process change models is created for the development of the conceptual model. The BPR model of Kettinger & Grover (1995) is taken as structure of the model and for the focus areas six important categories are identified based on analyzing the components of multiple models: process, people, organizational structures & cultures, information, technology and the change management (Figure 16). Based on the literature, these categories seem to be the most relevant regarding a DevOps implementation. For each category several aspects of DevOps are identified based on literature (Figure 18). This conceptual model based on the literature can be seen as a first design for a general DevOps implementation model.

Sub question 3: What is the value of the theoretical design for large IT service organizations based on empirical research and expert knowledge?

In order to make the theoretical model more specific for the implementation of DevOps at large IT service organizations, empirical data is retrieved by conducting interviews with IT managers at nine case organizations. First, it was confirmed that there is no common DevOps definition, although the high involvement of the cultural aspect was recognized by almost all managers. Next to that, it was acknowledged that the identified categories are all important, but the main focus at the start has to be given to people and the change management, instead of technology which is often done. Further, the interviews provided overall insights in the importance of each aspect per category for IT service organizations. This resulted in an indication of aspects that are relevant for DevOps, aspects that would be nice to have, aspects that already implemented due to an Agile way of working and aspects that are not DevOps specific within large IT service organizations (Table 5). It also provided other additional findings on the DevOps interpretation and implementation (Figure 37). Most additional findings were evaluated on its correctness by an evaluation session with experts of Accenture. Unfortunately, not all additional findings are evaluated due to research limitations, which is also indicated in the model. Based on the analysis of the evaluation session, conclusions on the additional aspects were drawn and the aspects were added to the overview of relevant or nice-to-have aspects within the categories (Table 11) and also other aspects that related to the implementation (Table 10) are adopted. With these conclusions the theoretical model is complemented with specific insights on the implementation of DevOps at large IT service organizations. This contribution is used within the next section of the Design & Development phase to develop a final implementation model for large IT service organizations.

Sub question 4: How does a model look like for implementing DevOps in IT service organizations and how can this contribute to the implementation of DevOps?

For this sub question, the conclusions upon the combination of the theoretical model and empirical results are used to create a final implementation model. It is concluded that the dynamic characteristics of both the relevant aspects and their implementation approach are not described within the current literature models for business process change. Therefore, it is decided to create building blocks for the development of the DevOps implementation model, specifically for IT service organizations, that is based on elements of BPC models together with the empirical evidence from the interviews. This resulted in five building blocks which are 1) External factors, 2) Components and its aspects, 3) Implementation approach and 4) Organizational factors. The building blocks are combined in a final implementation model that exists of one table (Table 17) and one visualization (Figure 49). As the whole implementation is very complex and dynamic, the visualization of the implementation is simplified and additional comments are given together with the models, related to working Agile before implementing a DevOps way of working, the start focus of the components, the component curves and their spans of

control, radical & continuous approaches with exits and the pilot & the type of service. Next to that, an assessment tool is associated to the implementation model. The use of the model is illustrated with multiple use cases that indicate each a different organizational situation. This illustrates that the model can help an IT manager to indicate the relevant focus areas and aspects of DevOps for its implementation within large IT service organization with each its organizational factors. Also relevant aspects regarding the implementation approach that the IT manager has to follow can be taken from the model.

Based on the answers to the sub questions, the main research question can be answered:

How does a guiding model look like to support the implementation of DevOps in IT Service organizations?

This question is based on the research objective that is formulated in the beginning of the research: “*To develop a DevOps implementation model that can support IT managers of large IT service organizations, which have adopted an Agile way of working, with their process of implementing DevOps.*” The design of the implementation model satisfies this objective as it indicates the most relevant areas and aspects to focus on for a DevOps implementation at large IT service organizations. Next to that, relevant implementation aspects are given by the model that have to be taken into account.

The implementation model consists of a table (Table 17), a visualization (Figure 49) and additional comments on characteristic that were too complex to visualize (Page 102). The model can contribute to the implementation of DevOps for the IT manager of an IT service organization in multiple ways. The model indicates the relevant aspects of DevOps to focus on in common for IT service organizations, which the IT manager can use to decide on which aspect he is going to focus his DevOps implementation. The model identifies also the organizational factors that influence the degree on which DevOps can be implemented.

The elements of the BPC models are still visible, although the model is also made more dynamic by including the learning aspect with the maturity curves and the varying approaches of implementation. The external drivers show the aspects that can be the reason to implement DevOps. The components, people, process, information, organization structure & culture, technology and change management are less visible in the complete model, although they are very important and described in detail in Table 17, associated to Figure 49. Each component has its own maturity curve per span of control, in which change management is indicated over each span of control. In this way the dynamic and adaptive focus areas are indicated. Learning by doing is key for DevOps as people will only learn on-the-job how to handle the DevOps way of working. Support and commitment of higher management is less visible, but is crucial for the success of the implementation of DevOps at a large IT service organization. The implementation is started at low level and spread over an expanding area within the business line. Between the different spans of control, radical changes are necessary in order to gain more mature in DevOps according to the model. However, it should be noticed that before a radical change is done towards a next span of control, the components at other levels can already be improved on continuous basis. The exits are not made visible in the model as this could suggest that this has to be done whereas it is an option. Exits are here defined as moments at which the implementation of DevOps will be stopped as it does not deliver the desired results or the maximal degree or most suitable DevOps way of working is already implemented. At such a moment it can respectively be decided to reverse the process back into the previous structure or keep the current DevOps way of working and not extend this anymore. In relation, not all business departments do not have to be integrated with the DevOps way of working. Finally, the organizational factors are influencing the whole implementation and have to be taken into account to create an organization specific DevOps situation.

The model can support the IT manager of a large IT service organization, which have an Agile way of working in place, with the implementation of DevOps. From the point of view of an IT manager, the implementation of DevOps can be guided by having this high level model to indicate the specific focus areas and aspects. It also states a common approach for its implementation at large IT service organizations.

Associated comments of implementation model of Table 16 and Figure 46

- **Working Agile**
The model assumes organizations work already according to an Agile methodology. If this is not the case, the grey aspects in Table 16 can indicate which Agile practices are important to get in place first, as well as other possible Agile practices, before implementing DevOps.
- **Start focus**
The focus areas for the start of the implementation is not visible in the figure, only in Table 16. The common start focus is on the components people and the change management, after that the process and organizational culture & structures and finally, the information and technology.
- **Aspects**
The overview of relevant aspects is given in Table 16. The ranking of the relevant comments can be seen as a common ranking, general to large IT service organizations. Depending on the organizational factors, this ranking has to be made organization specific by the IT manager. Next to that, the aspects indicated as nice to have can be seen as aspects that large IT service organizations will not focus on in the first place, but implement it when it is possible and a high degree of the other aspects is already in place.
- **Component curves**
Each component has its own curve. The curves are formed on basis of the degree of implemented aspects of Table 16. Per component the list of aspects can be filled in and the total degree can be determined, which will affect the height of the curve per span of control. The way on which the curves are formed present the adoption process of DevOps aspects in each component. As the most important performance measurement is value, the focus should always be on adding business value. This has to be translated in different KPIs across the maturity of DevOps.
- **Radical & continuous approaches with exits**
Between the different spans of control, radical changes are necessary in order to gain more mature in DevOps according to the model. However, it should be noticed that before a radical change is done towards a next span of control, the components at other levels can already be improved on continuous basis. This is visualized by the overlapping circles. Next to this, there are during the whole process exits in which the implementation can be stopped. Due to the experimental and learning by doing process in which DevOps has to be implemented, it always has to be possible to stop. Otherwise, the blameful characteristic will limit this experimental process approach.
- **Pilot & type of services**
For the DevOps implementation at large IT service organizations a pilot is often started at an application based service of an organization. This is not visualized, although it is an important factor as such services suit best for DevOps. By starting at this part of the organization, the lessons learned can be used during the implementation in the rest of the teams and business lines. The areas in which DevOps will be implemented is depending on the type of service that is being delivered by the business lines. Often a certain degree of the infrastructure, operations and business departments are not integrated in the full DevOps way of working due to the absence of a beneficial business case for the transformation towards DevOps at those areas.

Table 17: Implementation model (table)

	Process	People	Information	Org. Culture & Structure	Technology	Change management
Start Focus	2	1	3	2	3	1
Relevant aspects	Continuous Integration	Experimenting without regret	Transparency of org. information	KPIs	Automation	Treat employees equally
	Continuous Delivery	Encourage learning culture	Shared information	Incentives	Virtualization	Feedback sessions
	Systems Thinking	Willingness	Monitoring	Change in organizational structure	Build new technology next to monolithic system and replace	Create understanding
	End-to-end responsibility within self-organizing teams	Cross-skilled people	Standards for coding	Top management support & commitment	Design for Failure	Continuous improvement
	Focus on value	Sharing knowledge		Culture of collaboration	Continuous Deployment	Show results
	ITIL processes in Dev processes	Proactive mindset and attitude at the people		Budget, time and room to move towards new way of working	Focus on MVP	Training
	Balance between dev, innovation and problems	Different type of persons, based on its skills set which should include a basic and broad understanding of multiple areas		Environmental factors have to be adjusted as well, create a more creative and open place		Assign group
	Conflicting processes have to be aligned within the DevOps teams	Commitment of people involved		The KPI 'value' has to be translated in different metrics at different stages		Make an overview of automation
	Sprints	Cross- functional teams				People understand the reason of the change
	Feedback loops					

1 t/m 3 Ranking in start focus for implementation of DevOps

Relevant for DevOps

Relevant for DevOps (Not evaluated by Experts)

Nice-to-have

Implemented with Agile

Figure 48: Legend for Table 17

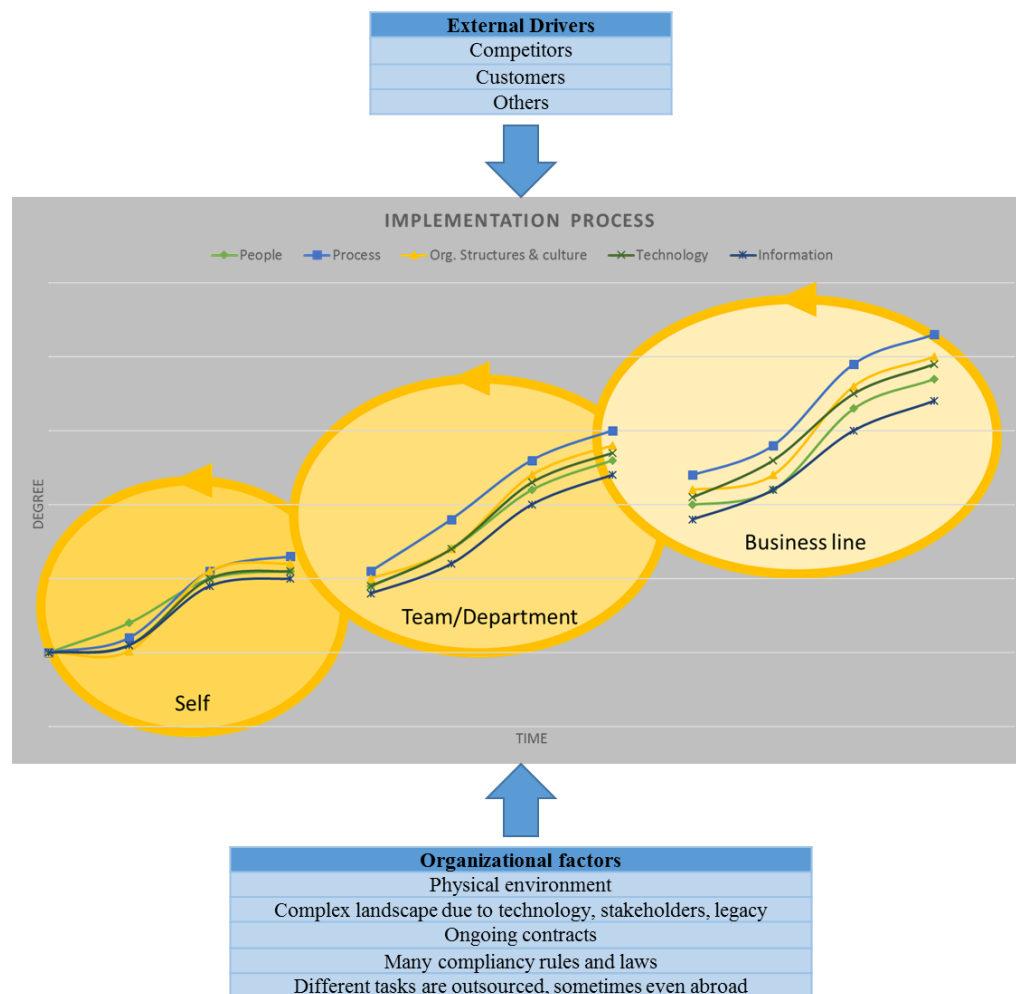


Figure 49: Implementation model (visualization)

6.2.1. Scientific Contribution

First of all, the research contributes to the limited amount of literature on DevOps with developing a well-defined DevOps definition. Also the differences of DevOps in relation to traditional and agile methodologies are defined and contributes to undefined characteristic of DevOps in literature. Next to that, the research is focusing mainly on changing the business process related to DevOps. Therefore, it is contributing to the scientific field of business process change models in relation with IT service development and delivery. The research provides a new insight on applying a business process change methodology to a transition of a DevOps way of working within IT service organizations.

The research approaches a DevOps implementation at large IT service organization as a business process change. Theory on business process change models have been examined in order to find a suitable conceptual model for this purpose. A combination of BPR and LSS is used to research the implementation of DevOps as this creates a model with a focus on both revolutionary and evolutionary changes. BPR as its clear structure on a business process change and several elements of LSS are already implemented in DevOps, but also the continuous improvement aspect is taken within the conceptual model. The final implementation model that is developed in this research still shows the elements that are designed in the conceptual model prior to the empirical research. The discussed reasons to take a top-down and bottom-up approach appeared to be valid. Also the components of the model are similar to the BPC models, although a combination has been made based on the most relevant for DevOps. However, the research shows that certain factors have to be taken into account for implementation process, which makes the process more dynamic.

As these factors are mainly related to the implementation process, it is good to compare the implementation approaches of the selected models in more detail.

Steps	BPR	Lean	Six Sigma
1	Prepare for BPR	Identify value	Define
2	Map & analysis as-is process	Map value stream	Measure
3	Design to-be processes	Create flow	Analyze
4	Implement reengineered process	Establish pull	Improve
5	Improve continuously	Seek perfection	Control

Table 18: Overview of implementation steps of BPC models (Kettinger, Teng, & Guha, 1997; Nave, 2002)

Based on the characteristics of DevOps and the research outcomes, DevOps relates the most to a combination of Lean Six Sigma. DevOps is for example focused on value streams within the business and seeks for improvements by monitoring and measuring. The process is continuously improved, without having the exact to-be situation analyzed first. However, the radical approach of BPR is at certain moments also needed as large changes within the organizational structure for example cannot be done continuously. Working according to DevOps desires some disruptive change within the organizational structure, but also regarding applying new technology or tooling for example. Therefore, a combination of both is needed. This combination is not straight-forward for the implementation of DevOps. DevOps desires an experimental learning process and this should also be adopted in the implementation process. Together with paying attention to the organizational factors, the implementation process has to be adaptive to its complex context. Within this research, this is done by taking the above models as basis and extend the models towards a dynamic application by adding the different factors. Therefore, the approach described in this model could be seen as an extension on the BPC models to make it more dynamic.

6.2.2. Societal Contribution

The described use cases of section 5.5 show illustrations on how the model could guide different IT managers during the implementation. The implementation model states that the focus and implementation of DevOps is situational due to multiple organizational factors. The organizational factors defined in the model are used to formulate different use cases, which are inspired by the cases that are used within this research. Each use case describes an organizational situation in which the model can support the implementation of DevOps. However, as multiple spans of control are used within the model, other roles within a large IT service organization could also use it to understand the focus areas or indicate where they are now as a team or complete organization. The model could already learn people that DevOps is not only about technology or only about culture, it is a mix of components that could be implemented based on choice and organizational factors. As the model is developed as general model, organizations within different sectors could use it as high level guidance. Based on the insights from the model, they can structure their DevOps implementation towards a suitable form for their organization. Future research could investigate what the major differences are between different sectors as during this research the organizational factors are already indicated as highly dependent on the implementation of DevOps.

Accenture can use this model to indicate on which areas and aspects of DevOps they have to focus the implementation of DevOps at their clients. Based on the insights of the model, more detailed operational steps on the implementation process can be defined. In fact, Accenture could also use this model in combination with a more operational maturity model of DevOps that is currently being developed. Both models could complement each other as the developed implementation model gives a high level view on the focus and approach during the implementation and the operational model describes operational steps that have to be undertaken during the implementation.

6.3. Limitations

The research has also certain limitations that are important to mention and discuss. The discussion of the limitations of a research can clarify the founded results and conclusions. Identifying limitations leads often also in opportunities for future research, which are discussed in section 6.4. The limitations can be categorized in limitations related to the research approach and scope (section 6.3.1), research methods (section 6.3.2) and the selection of experts and cases (section 6.3.3).

6.3.1. Research approach and scope

First, the limitations to the research approach and scope are discussed. The research is approached according to an adapted version of the Design Science Research framework of Peffers et al. (2007). Only the first three phases of the framework have been performed unfortunately, due to research constraints as time and resources. It could also be questioned whether this framework has also its own limitations and whether it is the most appropriate to use.

The design cycle framework of Peffers is structured with a sequential process with clearly defined stages. The framework initiates a process iteration, although only at the last two stages, being evaluation and communication (Peffers et al. 2007). For the phase Design & Development, it prescribes sequential phases with no feedback possibilities. Peffers' framework is partly based on the framework of Hevner et al. (2004) which prescribes a more iterative design process, however it did not adopt this iteration. For a research on DevOps, which aims to apply more agility within a development process, this iterative characteristic seems logical. Therefore, it can be questioned whether for this research the framework of Hevner would better fit. This can also be argued by the fact that within this research the design and development phase is split in three separate section and the evaluation step that is used, although such step in the development is not described in the design cycle of Peffers' framework. The model of Hevner describes the design cycle that makes use of a relevance cycle and the rigor cycle (Hevner et al. 2004). Within this design cycle the iterative process is defined. The application of Hevner's framework to the research could be started with the relevance of researching the DevOps implementing regarding the business field and is analyzed by conducting expert interviews. The following design rigor cycle relates the identified challenges of DevOps to literature on business process change models. The outcome of this literature research is evaluated on the application relevance by conducting case interviews which resulted in confirmations and additional information. This additional information is partly tested on its relevance correctness within the business field with a different research method, through an expert evaluation session. The drawn conclusions on the outcomes of the relevance cycle are reflected within the rigor that has been established through the earlier described literature research. Finally, the model is developed based on these conclusions and reflected upon its relevance and rigor contribution. Although the research is more based on the relevance cycle with its research methods then on the rigor cycle and Hevner's framework is also not completely followed within the research, the iterative nature of Hevner's framework seems to connect better to the research on the implementation of DevOps.

Next to this research approach, the chosen scope can be analyzed. The problem owner of the research is an IT manager of a large IT organization. Although it is acknowledged in literature that the IT manager plays a large role regarding the implementation of DevOps, it can be questioned whether this is the right scope to perform the research. Based on the research outcome it can be said that DevOps impacts almost the entire organization. It could be interesting to take the scope a level higher to higher management and involve business management as well. However, zooming in on the scope will identify the perspective of the people within the lower teams. Looking from this point of view, this research has taken a high level approach.

6.3.2. Research methods

Secondly, the research methods are critically analyzed. The research makes use of literature research, qualitative research in the form of expert and case interviews and an expert evaluation. Literature research is done based on much grey literature. As DevOps is a relatively new concept within the academic literature, it was hard to find well-established researches or literature on DevOps, besides

some more practical books. Although it was desired to have more grounded literature, avoiding the influence of grey literature on the research was impossible.

During the case interviews some quantitative data is gathered next to the qualitative data. Although this is analyzed and used within the research, the researcher argues that this can be seen as closed questions during the interview. Due to the fact that it is a small dataset and thus no significant results can be concluded, it is only used to gain a clear overview in the shared comments of the interviewees. It is acknowledged that this data cannot be interpreted as highly reliable data as the set is quite low. Together with the transcribed interviews it is still possible to use the data. A semi-structured approach is used for both the expert interviews as well as the case interviews. For both types of interview, a specific interview protocol is used. The results from the interviews are due to the protocols well-structured and this made the results more reliable.

The research makes use of different cases by interviewing one or more persons of an organization, nevertheless do not provided an in-depth case study on the involved organizations. Based on these interviews, the cases cannot be seen as full case studies, but remain case interviews. This is a fair limitation of the research as the interviews provided clear input for developing a general conceptual model. Though, not performing full case studies creates opportunities for future research. Related to this, the final model is unfortunately not evaluated or validated by another expert round or real case study. Due to time limitations within the research, this is not performed. However, this is also another future research aspect: validate the developed implementation model and continue improving or extending it.

Selection of experts and cases

As third, the selection of experts and cases can be reflected on. Both selections of the experts and the interviewees have been done by carefully formulated criteria. However, there are always certain limitations regarding this selection and the gained data from people as they can have an influenced perspective, for example due to the organization at which they work.

Another limitation that should be mentioned is that working with Accenture experts and clients creates a bias in the expert data and empirical data. Accenture experts are influencing the research with their point of views. Also selecting organizations from their clients creates a limited selection of organizations for the research. This is not unfair due to the collaboration of Accenture and their interest in the research outcome. Related to the selection of the organizations, organizations from multiple (private) sectors are involved in the research, while sometimes more from one sector and none from another sector due to availability and the Agile/DevOps way of working of the organizations. This is done to create the possibility to develop a general model as possible. The selection of the organizations is also done according to formulated criteria.

6.4. Reflection

Next to the limitations, a reflection can be done on the choices that are made within the research (Section 6.4.1) and on the relation with the master program for which this research has been done (Section 6.4.2).

6.4.1. Choices within the Research

First, the development of the theoretical conceptual model influences the research in a high degree. A business process change approach is selected within the research and this leads the research towards a certain direction. This choice is explained by stating that DevOps takes the whole business process into account. The empirical data shows that not all elements are covered within the currently existing models, while this could also be a result of combining the existing frameworks. Looking back on the research it may have been better to focus more on the implementation models instead of the components. Nevertheless, there is no consensus on what is all covered by DevOps, which makes it hard to focus directly on its implementation. Therefore, this model develops a research specific definition of DevOps and takes a business process change perspective that is being applied during the research and in the case interviews to gather the relevant focus areas for large IT service organizations taking this perspective as starting point. The business process change approach results in valuable insights, though it could also

be researched whether a system dynamics approach results in extra useful outcomes. This research indicated the important aspects and taking a system dynamics approach could indicate the influence and impact of the aspects on the complex socio-technical environment in more detail.

The use of the model for other implementation purposes can be reflected as well. This can be done according to an analysis of the building blocks. The building blocks that are formulated within the research are related to external drivers, components & aspects, implementation approach and organizational factors. The external drivers to implement a different concept could be the same as defined here. Many new concepts that emerge to be implemented within a business process are driven by the competitive environment and the external customer. The possibility of other drivers is also acknowledged within the system. In contrast, the other building blocks are more specific for DevOps. The components that are used in the model could be the same for the implementation of another concept, although research is needed to indicate the real relevance of these components in relation to the concept as well as extra components could be relevant that are not taken into account for the implementation of DevOps. The defined aspects of the components are highly depended on the concept that is being implemented. Within the six categories the aspects are specifically defined for DevOps, although most aspects of the change management component could be referred as general change management concepts. Therefore, these are also likely to be relevant for the implementation of another concept. Similar conclusion can be drawn for the component implementation approach, as for example starting with a pilot can be a suitable for other concepts as well. The organizational factors can also be used to identify the organizational barriers to implement a new concept. However, future research can be done on the actual value of the model to the implementation of another concept.

Reflecting on the research method, it could be questioned whether quantitative research, instead of qualitative research, would have resulted in another outcome. However, quantitative research on an implementation process requires that the process and its important factors are known. This research indicated the aspects on which such an implementation should be focused on. Based on these outcomes, quantitative research on these aspects could be done subsequently.

Moreover, as DevOps is approached as a way of working, it was good to perform interviews and do some observations to gain more knowledge on the actual way of working. As an example, when only an online survey was sent out to the interviewees, without personal visits of the researcher, the importance of the physical environment may not have been discovered. This was seen during the interviews and observations and otherwise such a question may not have been asked.

As stated before, the involved organizations were selected based on their size, their current way of working, their aim to implement DevOps and the availability of the interviewees. As the objective of the research was to develop a general model, the specific sector did not matter as long as organizations within multiple sectors were included. However, other options would be to focus the research on the implementation of DevOps within one sector or focus the research on small organizations. In the beginning of the research it has been said that the implementation of DevOps seems easier as smaller organizations often do not have much legacy. It can be interesting to focus a similar research on smaller organizations and compare it with the findings of this research. With focusing a similar research within one specific sector, the implementation model would become sector specific and typical organizational factors for this sector could be researched in more detail. Such research creates the opportunity to focus on the individual differences of the organizations within one sector, for example the current way of working which can be traditional or already full Agile. Next to that, due to the fact that DevOps is defined as an extension of Agile in this research, it is approached as that DevOps should be implemented in an Agile environment. This choice can indeed be questioned and it can be discussed whether it is not possible within another environment. However, for this research this choice has been made based on the definition of DevOps, being an extension of Agile. Future research could be an analysis of the influence different types of the current way of working.

6.4.2. Relation of research with SEPAM Master Program

Finally, the relation of the research with the master SEPAM-IA is reflected. The research was performed in the context of the SEPAM-IA master. The SEPAM master program focusses on designing solutions in complex socio-technical environments. With designing innovative solutions within socio-technical environments both the social aspects and technical aspects are important and have to be taken into account in the design. The implementation of DevOps results in different challenges regarding people, process and technology within an organization. This refers to a complex socio-technical environment which makes this research topic a relevant topic for the master program. Implementing DevOps can be seen as a multi actor system at different levels. There are different actors at team level, but also the different suppliers and partners are involved when the way of working is going to be changed. By taking multiple sectors into account in the research and develop a general model, it is avoided that the research is too much focused on one particular sector or organization. However, the research would have become more relevant for the SEPAM program when a distinction was made between the public and private sector. This is adopted as future research for this research. The IA track that is included in this master focusses on the alignment of organizational needs and the engineering opportunities and solutions offered by new IT solutions. DevOps is a new way of working for organizations with applying multiple technical innovations, which fits very well with the IA track.

Multiple courses of the master's curriculum contributed to the capabilities of the researcher to perform the research. First, the courses *Intro to Designing MAS (SPM4111)*, *Designing MAS from an Engineering Perspective (SPM4123)* and *Designing MAS from an Actor perspective (SPM4133)* helped to develop skills regarding defining the problem definition and objectives for a solution. By doing the projects of these courses, the researcher gained experience in tackling such problems within complex socio-technical environments. This experience increased by doing the course *MAS Design: An Integrated View (SPM4142)* in which elements of all three courses merged and a business process change project was analyzed from both technical and social perspective. Both the courses *Design of Innovative ICT-infrastructure and Services (SPM4340IA)* and *ICT Infrastructures Architectures (SPM4430)* created a knowledge base for the researcher in IT architectures and design for complex socio-technical systems. The course *Web Science & Engineering (IN4252)* prepared the researcher with writing an article with an IT topic, where the course *SEPAM Thesis Project Definition (SPM5905)* prepared the researcher to the entire process of performing a thesis project. The overall skillset that is part of the master program helped the researcher with the activities, such as presenting, writing and reasoning, during this project.

6.5. Future Research

Based on the limitations that are identified in section 6.3 and the reflections of section 6.4, it is possible to indicate interesting fields or gaps for future research. First, interesting aspects for future research that are not taken into the scope of this research are identified, where after associated research questions for possible future research are presented.

Following the limitation of the missing evaluation phase of the model, the eliminated phases of the framework of Peffers (2007) could be picked up as a follow-up study. Future research could do a more detailed practical demonstration, validate the value of the final implementation model and continue improving or extending it. Next to that, DevOps appears to operate in a complex and dynamic field with many stakeholders and complex changes. Based on this finding, another research approach could be taken with the focus on system dynamics for example. A system dynamics approach would create a different perspective on the implementation of DevOps by focusing on the non-linear behavior of the "actors" in the system. Such research could indicate the influence of different elements in the system related to DevOps and its implementation in a large IT service organization. Knowing the influence of multiple elements on the system and its environment can be an addition to this research in order to increase the knowledge on the implementation of DevOps.

The reflection on the used research method showed that qualitative research was the most appropriate to conduct as aspects on which an implementation should be focused on had to be defined first. Based on these outcomes, quantitative research on these aspects could be done subsequently.

An idea is to change the problem owner, as this would create another perspective. The research shows that a DevOps implementation touches almost the entire organization. Therefore, it can be interesting to take another perspective, that may be less obvious than de IT manager. On the other hand, the scope could also be zoomed in on the level of team members of the two departments to see in which way the interests are really conflicting.

Next to that, within this research the current way of working being Agile is used as starting point and criteria for the involved organizations. However, future research could take another approach and investigate the influence of different types of the current way of working to the implementation of DevOps.

The involved organizations are all operating within the private sector. It could be interesting to investigate the differences between private and public sectors, or how multiple organizations within one particular industry, with many compliancy laws for example, is dealing with DevOps. Future research can be done in order to make the model sector specific. Another option is to compare the model with an another similar research on the implementation of DevOps at smaller organizations. Next to that, it can be interesting to research the value of the model for the implementation of another concept as explained in the reflection.

Based on these recommendations the following possible future research questions can be formulated:

- What is the value of the final conceptual model in practice and how can this be validated?
- How could system dynamics support the implementation of DevOps?
- How could the importance of the indicated aspects to focus on be determined by quantitative research?
- What is the influence of the current way of working at an organization with focusing on one sector?
- Is there a difference in implementing DevOps between private and public sectors?
- How does the implementation of DevOps at smaller organizations relate to the findings on the implementation at large organizations?
- How can the implementation model be used for the implementation of other concepts, for example a new business process technology such as an ERP system, within the business process of a large IT service organization?

6.6. Recommendations

The research has been done with the involvement of Accenture as societal stakeholder. Their clients struggle with the implementation of DevOps and ask Accenture to help them with their approach towards DevOps. Although the model does not state extensive operational steps to follow for an organization, it provides a high level overview on the focus areas and its relevant aspects for the implementation of DevOps at large IT service organizations. It is recommended to take this model with its insights on the implementation of DevOps at large IT service organizations into account during analyzing the possible DevOps opportunities for clients. The assessment tool that is provided in this research could be used during the analysis of the current situation at client. It can be a first mean to see whether everyone's perspective in the organization is at the same page and what the current situation regarding DevOps is.

As recommendation for next steps to perform, the model could be extended. The relevant aspects could be developed in more detail with describing the operational actions that are required for the implementation of DevOps at large IT service organizations. The aspects are still at high level, but when the current situation and focus points of an organization are determined, the aspects can be extended with more concrete and practical steps to undertake. This could help to handle implications that are faced at a lower level regarding the DevOps implementation at large IT service organizations. The developed implementation model can also have a contributing value regarding a maturity model that is being developed at Accenture. The maturity model is quite operational and this implementation model is focused on a higher level, which could mean that both model could complement each other in its strengths.

References

References

- Abbadi, I. (2011). Self-managed services conceptual model in trustworthy clouds' infrastructure. *Workshop on Cryptography and Security in Clouds*, 1–10.
- Ambler, S. W., & Lines, M. (2012). *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press.
- Attaran, M. (2004). Exploring the relationship between information technology and business process reengineering. *Information & Management*, 41(5), 585–596. [http://doi.org/10.1016/S0378-7206\(03\)00098-3](http://doi.org/10.1016/S0378-7206(03)00098-3)
- Bang, S. K., Chung, S., Choh, Y., & Dupuis, M. (2013). A grounded theory analysis of modern web applications. *The 2Nd Annual Conference*, (October 2013), 61–62. <http://doi.org/10.1145/2512209.2512229>
- Barlow, J., Giboney, J. S., Keith, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., & Vance, A. (2011). Overview and Guidance on Agile Development in Large Organizations. *Communications of the Association for Information Systems*, 29, 25–44. <http://doi.org/10.2139/ssrn.1909431>
- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology*, 2(5), 2049–3444. <http://doi.org/10.15680/ijirce.2015.0305013>
- Beck, K. et al. (2001). Manifesto for Agile Software Development. *Agile Alliance*.
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30–39. <http://doi.org/10.1109/MS.2005.129>
- Boehm, B. W. (1988). Spiral Model of Software Development and Enhancement. *IEEE Internet Computing*. <http://doi.org/10.1109/2.59>
- Caum, C. (2013). Continuous Delivery Vs. Continuous Deployment: What's the Diff? Retrieved from <https://puppetlabs.com/blog/continuous-delivery-vs-continuous-deployment-whats-diff>
- Colavita, F. (2016). Devops movement of enterprise agile breakdown silos, create collaboration, increase quality, and application speed. In *Communications in Computer and Information Science* (Vol. 422, pp. 203–213). http://doi.org/10.1007/978-3-319-27896-4_17
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329–354. <http://doi.org/10.1287/isre.1090.0236>
- Conboy, K., Coyle, S., & Wang, X. (2010). People over Process : Key Challenges in Agile Development. *IEEE Software*, 48–57.
- Cucoranu, I. C., Parwani, A. V., & Pantanowitz, L. (2014). Lean Six Sigma, 14. <http://doi.org/10.1007/978-1-4614-9581-9>
- Davis, J., & Daniels, K. (2015). *Effective DevOps*. O'Reilly Media, Inc.
- Debois, P. (2011). Devops: A Software Revolution in the Making? *Cutter IT Journal*, 24(8), 1–41.
- Dingsoyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221. <http://doi.org/10.1016/j.jss.2012.02.033>
- Donaldson, L. (2006). The contingency theory of organizational design: challenges and opportunities. In *Organization Design* (Vol. 6, pp. 19–40). http://doi.org/10.1007/0-387-34173-0_2

- Earl, M. J. (1994). The new and the old of business process redesign. *The Journal of Strategic Information Systems*, 3(1), 5–22. [http://doi.org/10.1016/0963-8687\(94\)90003-5](http://doi.org/10.1016/0963-8687(94)90003-5)
- Emidio, T. D., Dorton, D., & Duncan, E. (2015). Service innovation in a digital world. *McKinsey Quarterly*.
- Erich, F., Amrit, C., & Daneva, M. (2014). *Report: DevOps Literature Review*. <http://doi.org/10.1007/978-3-319-13835-0>
- Erich, F., Amrit, C., & Daneva, M. (2016). *Cooperation Between Software Development and Operations: A Literature Review*.
- Fitzgerald, B., & Hartnett, G. (2005). A Study of the Use of Agile Methods within Intel. *Business Agility and Information Technology Diffusion*, 187–202. http://doi.org/10.1007/0-387-25590-7_12
- Fotopoulos, C. B., & Psomas, E. L. (2009). The impact of “soft” and “hard” TQM elements on quality management results. *International Journal of Quality & Reliability Management*, 26(2), 150–163. <http://doi.org/10.1108/02656710910928798>
- Galup, S. D., Dattero, R., Quan, J. J., & Conger, S. (2009). An overview of IT service management. *Communications of the ACM*, 52(5), 124. <http://doi.org/10.1145/1506409.1506439>
- Goldstein, J. (1999). Emergence as a Construct: History and Issues. *Emergence*, 1(1), 49–72. http://doi.org/10.1207/s15327000em0101_4
- Griffin, A., & Hauser, J. R. (1991). The Voice of the Customer. *Massachusetts Institute of Technology*.
- Guha, S., Kettinger, W. J., & Teng, J. T. C. (1993). Business process reengineering: building a comprehensive methodology. *Information Systems Management*, 10(3), 13–22. <http://doi.org/10.1080/10580539308906939>
- Hammer, M., & Champy, J. (1993). *Business process re-engineering*. London: Nicholas Brealey.
- Hanna, A. Rance, S. (2011). ITIL Glossary and Abbreviations. *ITIL Glossary and Abbreviations*, 1–93.
- Hassan, Q. F. (2011). Demystifying Cloud Security. *CrossTalk*, 16–21.
- Henderson, J. C., & Venkatraman, N. (1999). Strategic alignment: Leveraging information technology for transforming organizations. *IBM SYSTEMS JOURNAL*, 38(1), 472–484.
- Hevner, a. R., March, S. T., & Park, J. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–105. <http://doi.org/10.2307/25148625>
- Hicks, B. J. (2007). Lean information management: Understanding and eliminating waste. *International Journal of Information Management*, 27(4), 233–249. <http://doi.org/10.1016/j.ijinfomgt.2006.12.001>
- Howard, R. (2015). *The Top 10 Strategic Technology Trends for Government in 2015 (Gartner)*.
- Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. *Continuous delivery*. <http://doi.org/10.1007/s13398-014-0173-7.2>
- Hüttermann, M. (2012). Building Blocks of DevOps. In *DevOps for Developers* (pp. 33–47).
- Iden, J., Tessem, B., & Päiväranta, 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. <http://doi.org/10.1016/j.infsof.2010.12.002>
- Ikonen, M. (2010). Exploring the Sources of Waste in Kanban Software Development Projects, 2010. <http://doi.org/10.1109/SEAA.2010.40>
- Johnstone, C., Pairaudeau, G., & Pettersson, J. A. (2011). Creativity, innovation and lean sigma: A controversial combination? *Drug Discovery Today*, 16(1–2), 50–57.

- <http://doi.org/10.1016/j.drudis.2010.11.005>
- Jurisch, M. C., Cuno, J., Palka, W., Wolf, P., & Krcmar, H. (2012). An Integrative Model of IT-Enabled Business Process Change: Causal Structures in Theory, Research and Practice. <http://doi.org/10.1109/HICSS.2012.105>
- Kettinger, W. J., & Grover, V. (1995). Special section: toward a theory of business process change management. *J. Manage. Inf. Syst.*, 12(1), 9–30. <http://doi.org/10.2307/40398151>
- Kettinger, W. J., Teng, J. T. C., & Guha, S. (1997). Business process change: A study of methodologies, techniques, and tools. *MIS Quarterly: Management Information Systems*, 21(1), 55–79. <http://doi.org/10.2307/249742>
- Kim, G. (2011). Top 11 Things You Need to Know About DevOps. *IT Revolution Press*.
- Kim, G., Behr, K., & Spafford, G. (2013). The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win, 345.
- Kniberg, H. (2009). Kanban vs Scrum. In *InfoQ* (Vol. 1, pp. 1–41).
- Kniberg, H., & Ivarsson, A. (2012). Scaling Agile @ Spotify - with Tribes, Squads, Chapters & Guilds, 1–14.
- Kristekova, Z., & Jurisch, M. C. (2012). Consolidating findings from business process change case studies using system dynamics : The example of employee morale. *Knowledge Management & E-Learning: An International Journal*, 4(4), 455–480.
- Larman, C., & Vodde, B. (2013). Scaling Agile Development - Large and Multisite Product Development with Large-Scale Scrum. *CrossTalk*, (May-June), 8–12.
- Love, P. E. D., Heng, L. I., Irani, Z., & Faniran, O. (2000). Total quality management and the learning organization: A dialogue for change in construction. *Construction Management and Economics*, 18(3), 321–331.
- Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of DevOps. *Lecture Notes in Business Information Processing. Agile Processes, in Software Engineering, and Extreme Programming: 16th International Conference, XP 2015 Helsinki, Finland, May 25-29, 2015 Proceedings*, 212, 212–217. <http://doi.org/10.1007/978-3-319-18612-2>
- Maes, R. (1999). A Generic Framework for Information Management. *PrimaVera Working Paper Series*, 99(3), 22. <http://doi.org/10.1111/j.1467-629X.1980.tb00220.x>
- Mayer, R., & Benjamin, P. (1995). *A framework and a suite of methods for business process reengineering*. Idea Group Publishing. <http://doi.org/10.4018/978-1-87828-929-2>
- McAuley, L., Pham, B., Tugwell, P., & Moher, D. (2000). Does the inclusion of grey literature influence estimates of intervention effectiveness reported in meta-analyses? *Lancet*, 356(9237), 1228–1231. [http://doi.org/10.1016/S0140-6736\(00\)02786-0](http://doi.org/10.1016/S0140-6736(00)02786-0)
- Melrose, J., Perroy, R., & Careas, S. (2015). *DevOps for Dummies. Statewide Agricultural Land Use Baseline 2015* (Vol. 1). <http://doi.org/10.1017/CBO9781107415324.004>
- Mohammed, N., Munassar, A., & Govardhan, A. (2010). A Comparison Between Five Models Of Software Engineering. *International Journal of Computer Science*, 7(5), 94–101. <http://doi.org/10.1.1.403.3201>
- Moran, J. W., & Brightman, B. K. (2000). Leading organizational change. *Journal of Workplace Learning*, 12(2), 66–74. <http://doi.org/10.1108/13665620010316226>
- Nantermé, P., & Daugherty, P. (2016). The Technology Vision 2016 - People First: The Primacy of People in a Digital Age (Accenture), 1–70.

- Nave, D. (2002). How To Compare Six Sigma, Lean and the Theory of Constraints: a framework for choosing what's best for your organization. *Quality Progress*, 35(3), 73–78.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *Communications of the ACM*, 48(2), 72–78. <http://doi.org/10.1145/1060710.1060712>
- Paasivaara, M., Lassenius, C., & Heikkilä, V. T. (2012). Inter-team coordination in large-scale globally distributed scrum: Do Scrum-of-Scrums really work? *Empirical Software Engineering and Measurement (ESEM)*, 2012 ACM-IEEE International Symposium on, 235–238. <http://doi.org/10.1145/2372251.2372294>
- Pais, M. (2012). DevOpsDays Italy. Retrieved from <https://www.infoq.com/news/2012/10/introduce-devops>
- Peffer, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <http://doi.org/10.2753/MIS0742-1222240302>
- Pepper, M. P. J., & Spedding, T. A. (2010). The evolution of lean Six Sigma. <http://doi.org/10.1108/02656711011014276>
- Pollard, C., & Cater-Steel, A. (2014). Justifications, strategies and critical success factors in successful util implementations in U.S. and Australian companies: An exploratory study. *Geoscience and Remote Sensing Symposium (IGARSS)*, 530(1), 1–5. <http://doi.org/10.1007/s13398-014-0173-7.2>
- Prodan, M., Prodan, A., & Purcarea, A. A. (2015). Three New Dimensions to People , Process , Technology Improvement Model, 481–490. <http://doi.org/10.1007/978-3-319-16486-1>
- Puppet Labs. (2015). 2015 State of DevOps Report. *Puppetlabs*, (877), 37. [http://doi.org/10.1016/S0022-3913\(12\)00047-9](http://doi.org/10.1016/S0022-3913(12)00047-9)
- Rouse, M. (2014). Two-speed IT Definition.
- Sabatier, P. A. . (1986). Top-down and Bottom-up Approaches to Implementation Research : A Critical Analysis and Suggested Synthesis. *Cambridge University Press Stable*, 6(1), 21–48.
- Salehi, F., & Yaghtin, A. (2015). Action Research Innovation Cycle: Lean Thinking as a Transformational System. *Procedia - Social and Behavioral Sciences*, 181, 293–302. <http://doi.org/10.1016/j.sbspro.2015.04.891>
- Sato, H. (1927). The blind men and the elephant. *Notes and Queries*. <http://doi.org/10.1093/nq/CLIII.dec10.425-a>
- Schaefer, A., Reichenbach, M., & Fey, D. (2013). Continuous integration and automation for DevOps. In *Lecture Notes in Electrical Engineering* (Vol. 170 LNEE, pp. 345–358). http://doi.org/10.1007/978-94-007-4786-9_28
- Schwaber, K. (1995). Scrum development process. *Proceedings of the Workshop on Business ...*, (April 1987), 10–19. http://doi.org/10.1007/978-1-4471-0947-1_11
- Schwaber, K., & Beedle, M. (2001). *Agile Software Development with Scrum*. *cdswebcernch* (Vol. 18). <http://doi.org/10.1109/2.947100>
- Serrat, O. (2010). Bridging Organizational Silos. *International Publications, Paper 120*(July), 1–5.
- Shahin, M. (2015). Architecting for DevOps and continuous deployment. *ACM International Conference Proceeding Series*, 28–Septemb, 147–148. <http://doi.org/10.1145/2811681.2824996>
- Staats, B. R., Brunner, D. J., & Upton, D. M. (2011). Lean principles, learning, and knowledge work: Evidence from a software services provider. *Journal of Operations Management*, 29(5), 376–390. <http://doi.org/10.1016/j.jom.2010.11.005>

- Swartout, P. (2014). *Continuous Delivery and DevOps: A Quickstart guide, Second Edition*. <http://doi.org/10.1017/CBO9781107415324.004>
- Ten Hagen, J., & Heunks, J. (2016). Inbedden van DevOps. *Informatie, Maandblad Voor Informatievoorziening*, 26–33.
- Tessem, B. ., & Iden, J. . (2008). Cooperation between developers and operations in software engineering projects. *Proceedings - International Conference on Software Engineering*, (January 2008), 105–108. <http://doi.org/10.1145/1370114.1370141>
- Van Bon, J. (2004). *IT Service Management an Introduction based on ITIL. ITSM Library*.
- Velasquez, N., Kim, G., Kersten, N., & Humble, J. (2014). 2014 State of DevOps Report.
- 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. <http://doi.org/10.1109/INTECH.2015.7173368>
- Walls, M. (2013). *Building a DevOps Culture. O'Reilly Media, Inc.*
- Weerakkody, V., Janssen, M., & Dwivedi, Y. K. (2011). Transformational change and business process reengineering (BPR): Lessons from the British and Dutch public sector. *Government Information Quarterly*, 28(3), 320–328. <http://doi.org/10.1016/j.giq.2010.07.010>
- Wells, C. J. (2014). Systems Analysis and Design - The Waterfall Model. Retrieved from http://www.technologyuk.net/computing/sad/waterfall_model.shtml
- Williams, S. R., & Baxter, J. a. (1996). Develop Long-Term Competitiveness through IT Assets. *The Elementary School Journal*, 97, 21–38. <http://doi.org/10.2307/256788>
- Womack, J. P., & Jones, D. T. (1996). *Lean Thinking. The Library Quarterly* (Vol. 5). <http://doi.org/10.1086/601582>
- Yu, X., & Petter, S. (2014). Understanding agile software development practices using shared mental models theory. *Information and Software Technology*, 56(8), 911–921. <http://doi.org/10.1016/j.infsof.2014.02.010>

Appendices

Appendices

Appendix A – Overview of Exploratory Expert Interviews

Overview of the exploratory interviews with Agile/DevOps experts

Interview	Expert	Role	Years of experience	Date of interview	Language
Interview 1	Exp1	Service Management Specialist	2 years	08-08-2016	Dutch
Interview 2	Exp2	Senior Manager Service Delivery	3-year with Agile/DevOps 12-year experience with app/infra delivery	18-08-2016	Dutch
Interview 3	Exp3	Technology Consulting Senior Manager	3 years	31-08-2016	Dutch
Interview 4	Exp4	Technology Consulting Manager	5 years	31-08-2016	Dutch
Interview 5	Exp5	Service Management Associate Manager	6 years	13-09-2016	Dutch
Interview 6	Exp6	Technology Architecture Senior Manager	4 years with DevOps 10 years with Agile	14-09-2016	Dutch
Interview 7	Exp7	Infrastructure Manager	5 years with DevOps 8 years with Agile	02-09-2016	English

Appendix B – Interview Guide Experts

Interview Guide Experts

Organization:		Operation Group:	
Interviewer:		Date:	
Interviewee(s):		Function:	
Accenture involvement:		Years in function:	

1. Personal introduction
 - a. Introducing myself
 - b. Ask interviewee to introduce himself
2. Introduction to research topic DevOps
 - a. Research objective
 - b. Research definition
3. DevOps characteristics / success factors
 - a. Experience cases of person
 - b. Why DevOps in each case? Objective → Strategy
 - c. Starting condition per case (ITIL/Agile)
 - d. Performance measurement
 - e. Project variables:
 - Dynamics of environment
 - Complexity
 - Size / Ambiguity
 - Uncertainties
4. Implementation DevOps
 - a. Requirements / Problems related to:
 - Management
 - Business processes
 - IT
 - Structure
 - People
5. Wrap-up
6. Follow-up

Appendix C – Interview Guide Case Interviews

Interview Guide

Organization:		Operation Group:	
Interviewer:		Date:	
Interviewee(s):		Function:	
Accenture involvement:		Years in function:	

1. Preparation

- a. Do you mind that I record this interview?

2. Personal introduction

- a. Introducing of myself: Who am I? Who is Accenture and how they are involved? What kind of information am I looking for? Why am I speaking to him/her specifically? s
- b. Ask interviewee to introduce himself: What is your position? What are your primary responsibilities? How long have you been at this organization or position?

3. Introduction to research topic

- a. For my thesis I am doing research on DevOps and its implementation at larger IT service organizations. The research question is: How does a guiding model look like to support the implementation of DevOps in IT Service organizations? For this purpose, I trying to discover what the current situation at those organizations is and which aspects of DevOps are the most relevant. Therefore, it is interesting to see how the process of business, development and operations looks like at this organization and what could or should be changed for a transition to DevOps.
- b. Before I start, could you tell me what is DevOps in your eyes?
- c. Introduction in research model:

DevOps Research

Research definition of DevOps:

"DevOps is a cultural movement that breaks silos between the business, development and operations departments, combined with a number of service development practices regarding people, process and technology, that enable rapid development and delivery of services"

Focus of research: IT manager or similar role(s)

Main focus on the 6 categories:

- Process
- People
- Information
- Organizational structure
- Technology
- Process / Change management approach

4. Current situation organization

Indication of current situation



Focus areas for change when implementing DevOps?

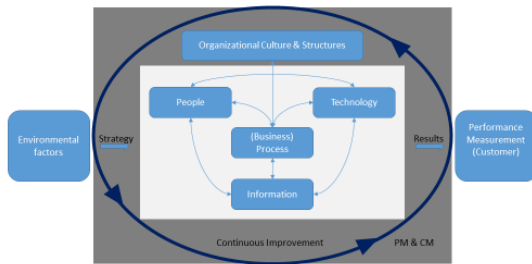


Open questions

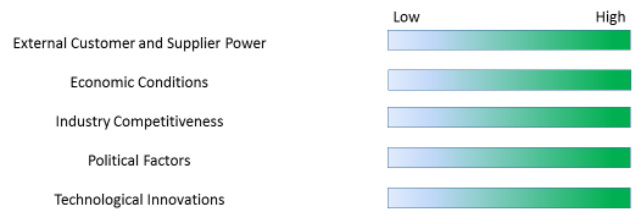
- a. How does the current pipeline of the process (business, development and operations) and the handovers between these departments looks like? What is outsourced?
- b. To what extent did IT project management methodologies change recently? How is this managed within the organization? How are the employees taking along with these changes? Are there trainings? Is there understanding under employees?
- c. How is the software development arranged? Is this according to the waterfall model or Agile? In which way is the Agile way of working implemented? Do you use Scrum? How is this arranged with employee contracts, KPIs, organizational structures?
- d. How is the operations department arranged? Is this according to the ITIL framework?
- e. How are handovers handled between departments handled?
- f. Why and to what extend could DevOps be beneficial for the organization?

Closed questions

Conceptual model



What would be/are the main reasons to adopt a DevOps WoW?



When implementing DevOps, which aspects would be/are the most important measurements to focus on?



Processes

When implementing DevOps, which aspects would be/are the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the importance of each aspect when implementing DevOps:				
	Yes	No	1	2	3	4	5	1	2	3	4	5
System thinking (The to-be delivered product is viewed holistic)	Yes	No										
End-to-end responsibility within self-organized teams	Yes	No										
Focus on customer (priorization of the requirements)	Yes	No										
Short iterations, sprints are used	Yes	No										
Feedback Loops, internal and external	Yes	No										
Continuous integration (discipline of building and validating a project automatically whenever a file is checked into the configuration management system)	Yes	No										
Continuous Delivery (on-demand releases)	Yes	No										
Ops (ITIL) processes integrated in Dev process	Yes	No										
Balance between product development, innovation and incident / problem management	Yes	No										
Remove non-value adding activities	Yes	No										

People

When implementing DevOps, which aspects would be/are the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the importance of each aspect when implementing DevOps:				
	Yes	No	1	2	3	4	5	1	2	3	4	5
Cross-functional teams (BAs, Developers, Testers, QA, Operational staff included)	Yes	No										
Cross-skilled people (Dev, test and Ops tasks are executed by everyone)	Yes	No										
Learning (encourage people to discover other department's tasks)	Yes	No										
Knowledge sharing (knowledge management among the team)	Yes	No										
Experimenting without regret (enable experimenting blameless culture)	Yes	No										
Cooperation: willingness to cooperate / work in new manner	Yes	No										

Information

When implementing DevOps, which aspects would be/are the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the importance of each aspect when implementing DevOps:				
	Yes	No	1	2	3	4	5	1	2	3	4	5
Transparency of organizational information	Yes	No										
Information shared and accessible throughout the process	Yes	No										
Standards in documentation and coding (one language)	Yes	No										
Monitoring of information and process	Yes	No										

Organizational culture & structure

When implementing DevOps, which aspects would be/are the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the importance of each aspect when implementing DevOps:				
	Yes	No	1	2	3	4	5	1	2	3	4	5
Change in organizational structure, with new Agile (Scrum) roles; Jobs and contracts based on Agile roles												
Incentive structures on team instead of individuals												
Top management commitment, support and facilitate process												
Appropriate KPIs for Agile (i.e. delivery pace, velocity or ...)												
Culture of collaboration; open communication, incentive and responsibility alignment, respect and trust												

Technology

When implementing DevOps, which aspects would be/are the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the importance of each aspect when implementing DevOps:				
	Yes	No	1	2	3	4	5	1	2	3	4	5
Continuous Deployment pipeline is present												
Automated activities (e.g. configuration, tests, releases)												
Virtualization & Cloud-based infrastructure and applications (IaaS, PaaS, SaaS, XaaS)												
Focus on minimal valuable product and extend with features												
Design for failure (not preventing failure, but designing resilient services in which inevitable failures have a minimal effect on service availability and functionality)												

Change Management

When implementing DevOps, which aspects would be/are the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the importance of each aspect when implementing DevOps:				
	Yes	No	1	2	3	4	5	1	2	3	4	5
Create understanding of Agile/DevOps WoW at employees												
Training of employees in handling the DevOps WoW / process												
Make an overview of the process pipeline (automation)												
Assign a central group / team to support transition to DevOps												
Treat employees of all departments equally												
Show results and make improvements visible												
Enable continuous improvement												
Create feedback sessions												

5. Implementation DevOpsOpen questions:

- ITIL/DevOps?
- DevOps is related to continuous delivery, however, it is also implying continuous deployment. Is it desirable for your organization to implement continuous deployment and why? According to the business needs? How relates this desire to IT limitations?
- What would prevent you from transforming to DevOps? What would be/are barriers, internal and external of the organization? Are there constraints that limit this transformation?

6. Wrap-up

- Is there anything I have forgotten to ask?
- Is there any secondary data such as documents, presentations, papers... that you could share with me on a confidential basis?
- Are there other interesting organizations/projects/cases that are either DevOps pioneers or facing adversaries in adopting DevOps? Are there any other interviewees that are interesting for me to speak with?

7. Follow-up

- Thank you
- Confirming e-mail with transcript of interview
- Follow-up e-mail

Appendix D – Questions Evaluation Session

#	Question	Answer options
1	What is DevOps according to you?	Free text
2	What would be your main focus area when transforming your organization to DevOps?	Process, People, Technology, Information, Org. Culture & Structure, Change management
3	You have to work Agile to implement DevOps	True/False
4	A special group or team is needed to guide the DevOps transformation	True/False
5	If you implement DevOps, there is no way back	True/False
6	DevOps (IaaS) have to be applied to all infrastructure of an organization	True/False
7	You have to hire different kind of people in a DevOps environment	True/False
8	Our clients need a complete reorganization to be able to work in a DevOps way of working	True/False
9	Large organizations have too many existing contracts and legacy to adopt DevOps	True/False
10	A DevOps transformation should be approached from bottom-up, in the following order: 1. Individual, 2. Team, 3. Organization	True/False
11	DevOps is not implemented in phases, but in a continuous experimental process	True/False
12	Organizations have to adopt to a FULL DevOps way of working. There is no such thing as half DevOps	True/False
13	The physical environment has to be adjusted in order to encourage a DevOps way of working	True/False
14	Different compliancy rules and laws are limiting organizations from adopting a DevOps way of working	True/False

Appendix E – Case Descriptions

6.6.1. Case A: Financial Services

The first case is based on an organization within the financial services industry that has undertaken quite some disrupting changes in adopting a new way of working. The interview was conducted with an IT manager, or so-called Chapter lead, within the Business Intelligence department. Next to the interview, a presentation on the organizational transition was used as secondary data for this case.

The organization has started their organizational transformation around 2010. More attention was given to the IT aspect within the organization and in 2015 their DevOps way of working was implemented in an early stage. The transformation started at the mobile department of the organization as *“this department lend oneself the most for DevOps, for example for continuous deployment, where after it is spread over the entire company”* (A1). First the departments, development and operations, were brought together in an informal way. This can be seen as the preparations for the big organizational change which was done with a big bang approach. Up to now, they are still busy with transforming their way of working into a more total DevOps way of working. This resulted in different maturities of DevOps between different squads where in the eyes of the interviewee none of the squads are fully DevOps yet, as far as this even exists. Although the organization calls their current way of working BusDevOps, the organization still exists of two main divisions, Business and IT. The next step will be to bring those divisions together and continue with one complete division.

6.6.2. Case B: Communications & High Tech

For case B an employee of Accenture was interviewed who is an Agile coach. He led the transformation at this smaller organization within the Communications & High Tech industry, which is a subsidiary organization of the organization of case C. Also a chain manager at the operations side of the subsidiary organization was being interviewed. This manager has worked previously for the larger organization, but has worked for quite some years at the subsidiary organization. Currently the manager works for the larger organization, but provided answers from both perspectives. These two interviews gave a clear overview of the way of working at a subsidiary organization of the larger organization.

According both interviewees, this subsidiary is a further stage regarding Agile/DevOps. As this organization is of smaller size and has less legacy than its larger organization, there are less barriers to implement a new way of working. The organization is younger, faster and more dynamic in its core and the decision making policy is less complex. Therefore, it is easier to focus on the important things for all stakeholders and the connection between development and operations was done quicker. The development department worked already according to an Agile manner, however, it was hard to get operational problems into the scope of development. According to the chain manager this has changed: *“When the understanding and collaboration became better though, the problems were easier shared and picked up faster”* (B2). The two departments are the organization are therefore more and more joint, but they are not working fully DevOps yet.

6.6.3. Case C: Communications & High Tech

Case C is describing the situation at an organization within the Communications & High Tech industry. The interview was done with an infrastructure consultant who has a background in operations and was also chain manager for two years before his current function. This is the same manager that has been chain manager for the subsidiary organization of the organization. In his current role he is working between the fields of development and operations. Also two other employees of Accenture with a supporting role in the transformation of this organization, each in a different department, have been interviewed. Also a presentation of a senior manager from Accenture is used to gather more information on the undertaken transformations at this organization.

The Agile way of working is already embraced within the organization, mainly at the development departments. In order to implement DevOps, an Agile way of working was first applied to the operations side. After that the connection with the development departments is tried by making the connection from

operations side. However, this is not yet done at all divisions of the organization. One division of the organization is quite successful in implementing DevOps in their way of working. The aspects listed below are the lessons learned at the transformation at one division of the organization, which is now currently working in a mature DevOps way. The idea is to use these lessons at the other departments and to enlarge the application of DevOps over the rest of the organization. However, it is acknowledged that this will be hard due to all the legacy systems and also the traditional culture of the organization.

Lessons learned:

- Invest in communication and active coaching
- Show results and participation
- Interface management and consolidation
- Intense coaching, breaking silos & business value focus
- Lean IT processes and automation
- Enforce continuous improvement and delivery value
- Dedicated service integration team
- Build relationship and cooperate in win-win situation
- Feedback loop production to DevOps teams
- Tooling service catalogue and supporting processes

6.6.4. Case D: Products

For the fourth case the way of working at a transport organization is analyzed. Two interviews were conducted with IT managers, one from development and one from operations. The IT manager on the development side has almost seven years' experience within the organization and in its current role three years. The IT operations manager has nine years of experience at the organization and is mainly focusing on the functional operational side.

Currently there are still silos from a hierarchical point of view. Both department are still separate, however, *“since a few months the functional operators are informally put together with the developers, as a first step towards DevOps”*. In this way the interest gap between the two department is already fading. The departments know better what is happening at the other side. However, this is only within the internet division of the organization. This phenomenon is less visible at the other divisions, though there are plans to implement DevOps there as well. This organization defines also a clear separation between Continuous Delivery and DevOps. Both interviewees state that the current focus is on Continuous Delivery with the right technology and a lower priority is given to DevOps. Although they both agree that it is closely intertwined, the separation makes it possible to focus on the right thing at the time.

6.6.5. Case E: Products

After the interviews for case D, the interviewees recommended to explore the mobile division of the organization as well, as this was a smaller subsidiary of the organization and works on an independent and Agile way. Therefore, case E is based on an interview with the manager of the mobile division. This manager can also be seen as the product owner of this department and the department can on its turn be seen as a small enterprise within the larger organization.

This mobile department has emerged from scratch within the organization and it is responsible for the national and international mobile applications of the organization. They started with a new way of working by themselves that nowadays can be called PostScrum. The team is being called 'blended' as each team member has multiple skills. They rotate very often their tasks in order to keep track of each other's work. The department is also working in the Cloud. All these aspects sound quite like a DevOps way of working and therefore it seems as they have gone through quite some changes and transformed their way of working. However, it has to be noticed that this department has arisen from scratch and did not have much legacy. In that way, they operate independent of the larger organization, although they keep dependent of the larger back office and work with APIs. All these characteristics made it easy to

adopt a DevOps way of working. In order to adopt this way of working also in the larger organization, many different teams in the rest of the organization are learning from this department.

6.6.6. Case F: Financial Services

Case F is an organization within the financial services operating group. The organization is currently changing their way of working more and more into a DevOps way of working. The interview was conducted with a domain manager with two years of experience, who is responsible for six Scrum teams at the development side. As secondary data, a presentation on the DevOps journey within this organization was used.

Although this organization is quite busy with investing time in a new way of working, they have extra difficulties on top of the legacy that all large organizations have. As they work with financial data of their customers, they have to ensure that they are compliant with certain rules and this includes some restrictions in the way of working. They are not sure if a DevOps way of working is applicable for the whole organization. However, some Agile way of working characteristics are already implemented and they are investigating how they can work in a more DevOps way. The organization is therefore not yet moving towards DevOps, but the intension is definitely present at the organization. The teams at the floor desire to work more DevOps, however, the larger overall organization hold them back.

6.6.7. Case G: Financial Services

Two interviews were held for this case of a financial services organization. Two technical architects were interviewed, of which one is part of the DevOps change team within the organization. Next to that, also some Accenture employees have given some interesting insights on the way of working at the organization, where one of them was an Agile coach that is supporting the transition of DevOps at this organization.

Within this organization, it is also very varied in terms of the current way of working. One of the interviewees said that the departments are minimal working according to Waterfall and as a maximum Agile on a mature level, with looking for DevOps possibilities. Another interviewee mentions that these Agile way of working is mostly visible at temporarily projects. Next to that they have a project with their DevOps change team in which they research and prepare the most appropriate way of DevOps for their organization. Within this project the implementation is approach from the technical perspective. The idea is to implement DevOps in the rest of the organization through this project. However, it is questioned by one interviewee whether this DevOps way of working is useful and necessary for all departments of the organization.

6.6.8. Case H: Communications & High Tech

This organization is supporting organizations with their services in the Communications & High Tech industry and the interview was conducted with a service manager of a department that is working for one particular large client. The service manager has four years' experience in this field. Next to this interview, also an observation of one day is being done. During this day, several meetings were attended, such as the sprint demo, sprint retrospective and the sprint planning.

The organization has their DevOps way of working mainly focused on the people aspect. This resulted in *“rotating people between functions in the two departments, in order to create a shared understanding and level of skills at all people involved, except for testing”* (H1). In some ways they are more depended on their customer in comparison with the organizations of the other cases as they work B2B. Their client can decide whether they want to invest in technology for example. That is why they basically focused their DevOps way of working transition on their people for themselves. During the process all Scrum / Kanban principles were visible and they are very well implemented in the mind-set as well: *“You can notice that developers are taking the operations part into account and they are for example also initiating a role change every now and then”* (H1).

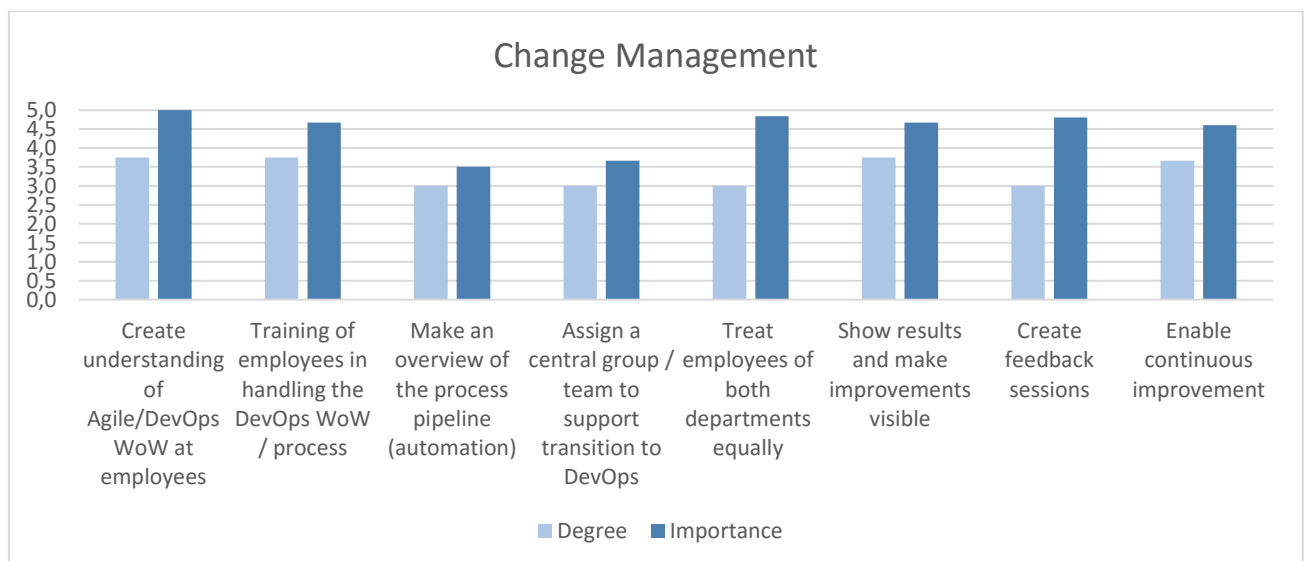
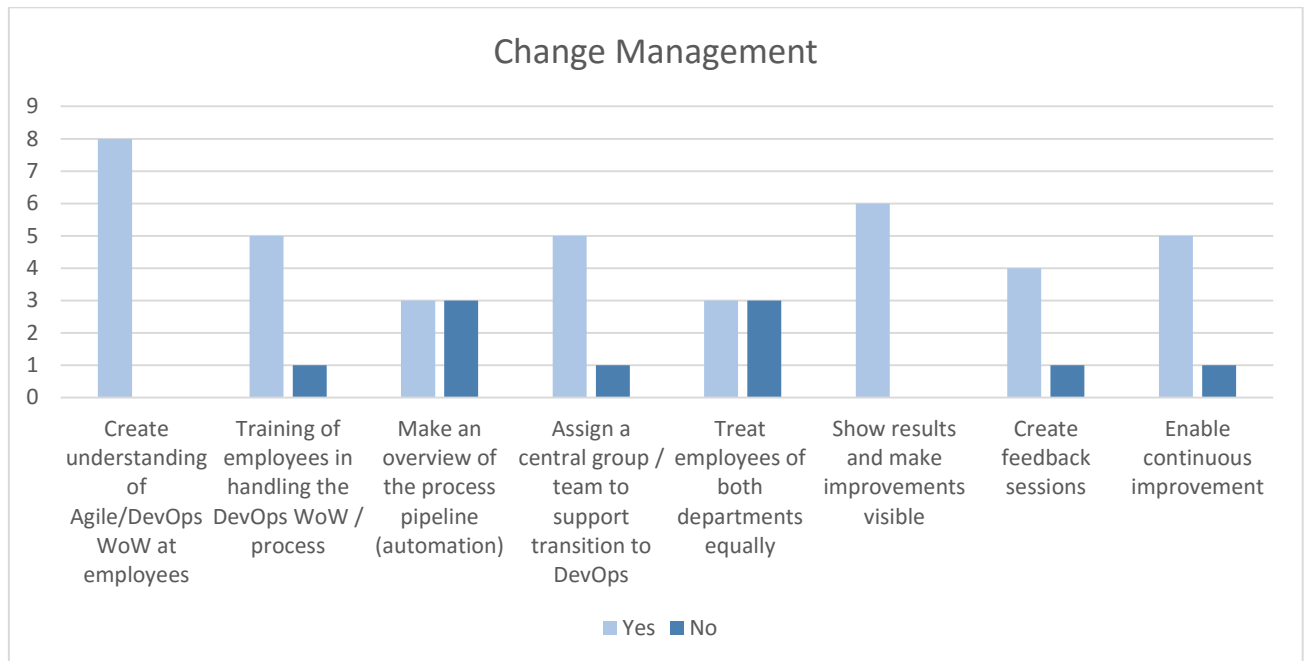
6.6.9. Case I: Products

The organization is a retail company and it has a large department for their online applications. The interviewee is a development manager and has the responsibility over all the external hired employees within the Scrum teams. The online department in which he works is already working with Scrum for over eight years.

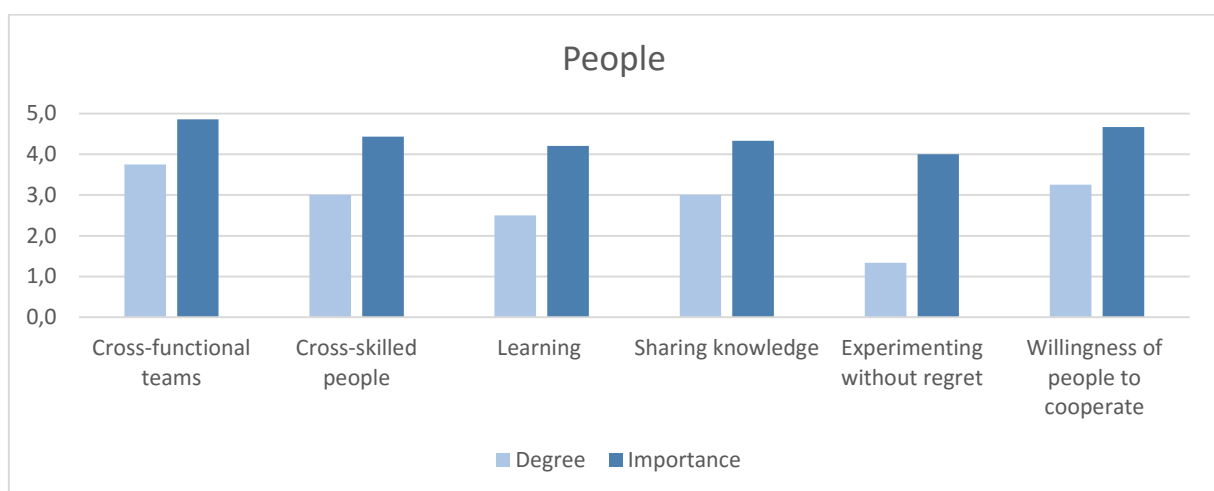
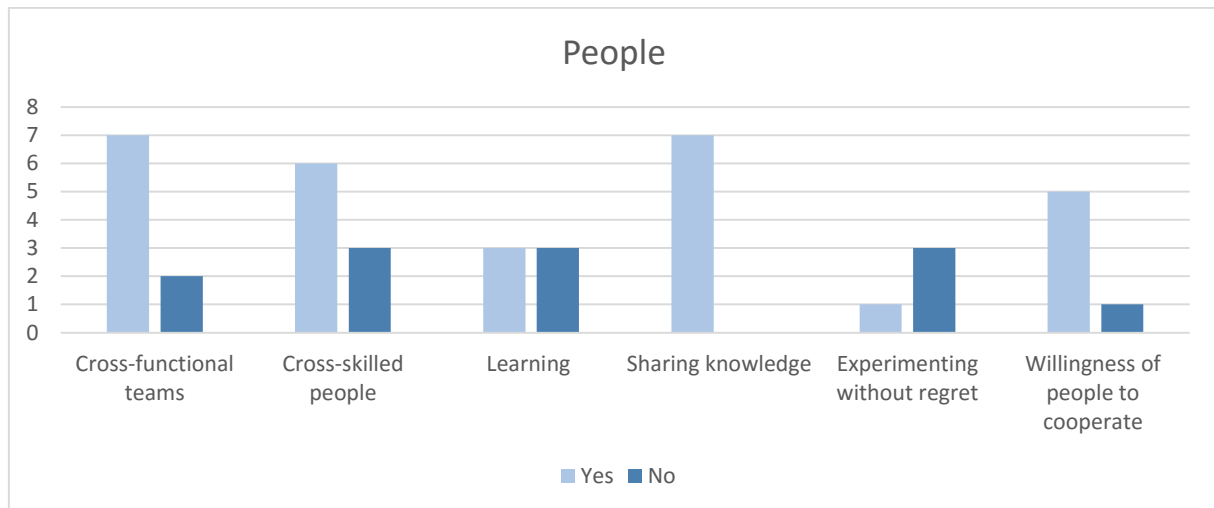
Within this department the transition towards a DevOps way of working is started about a year ago. They have undertaken already some changes, however the separate development and operations departments still exist although they have shared components. Their development department has grown over the last years and the amount of employees at their operational department remained the same. This makes it easier to create a DevOps way of working as operations already dedicated much of their tasks to development. This is different at other divisions of the organization. As the manager says: “*At other divisions is the separation between development and operations stronger as operations is a much bigger department with much knowledge and responsibilities*” (I1). The transition to full DevOps within this department, and in the rest of the organization even more, is limited by corporate characteristics that the organization has, as the interviewee argued.

Appendix F – Results Detailed Tables

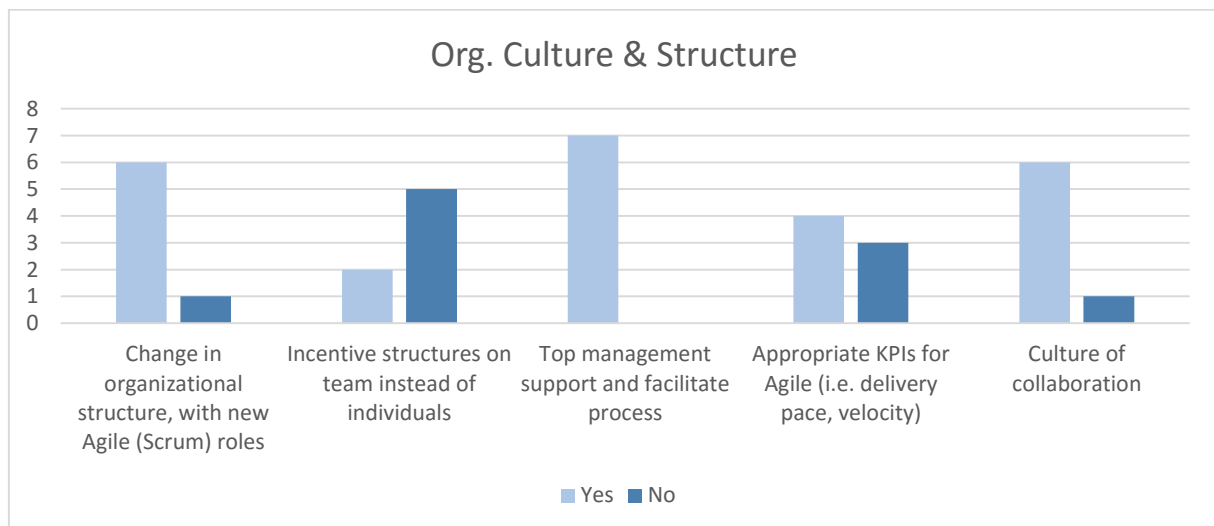
Process/change

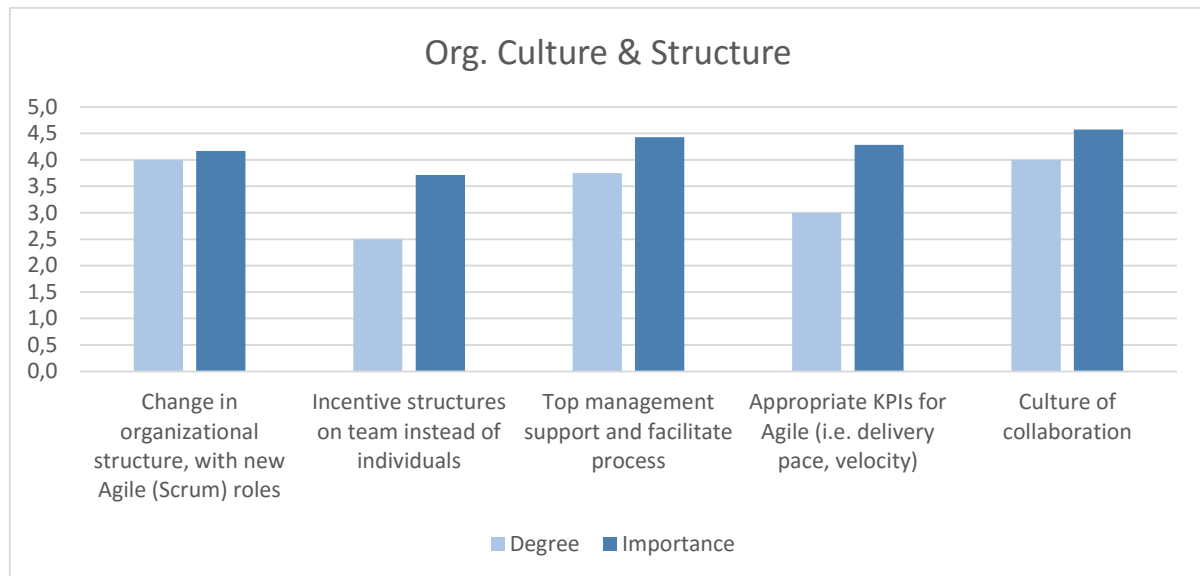


People

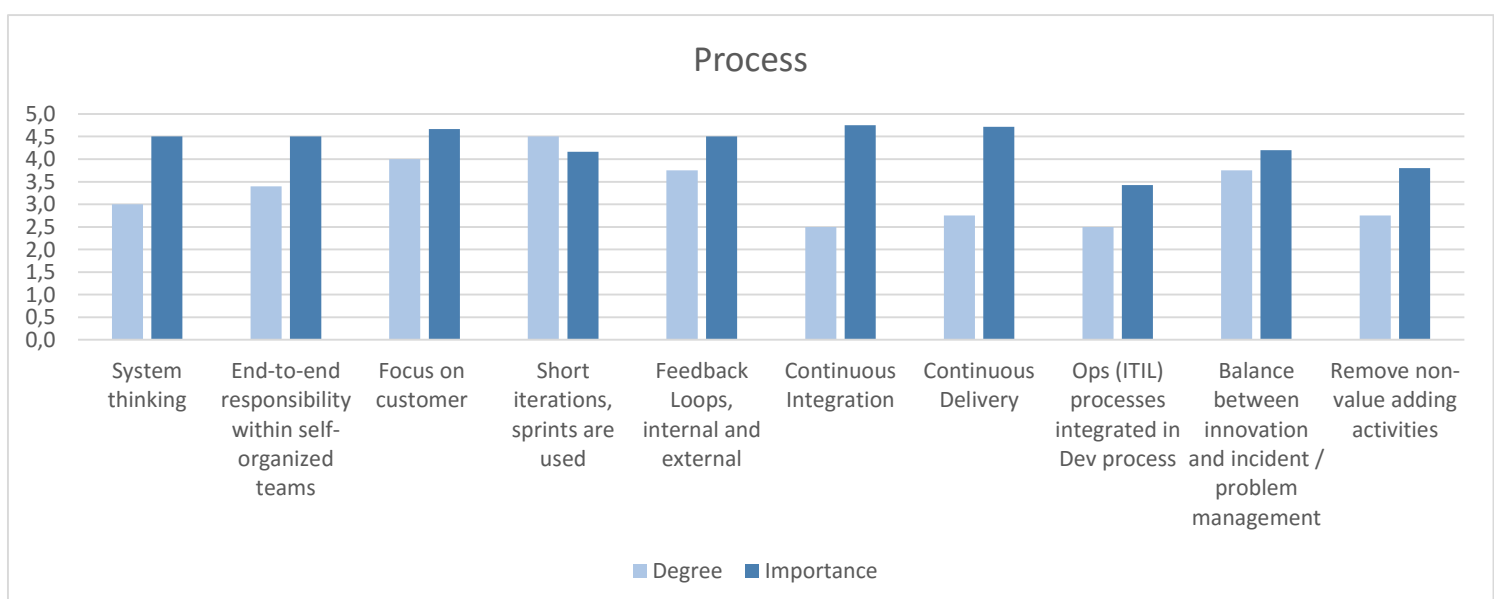
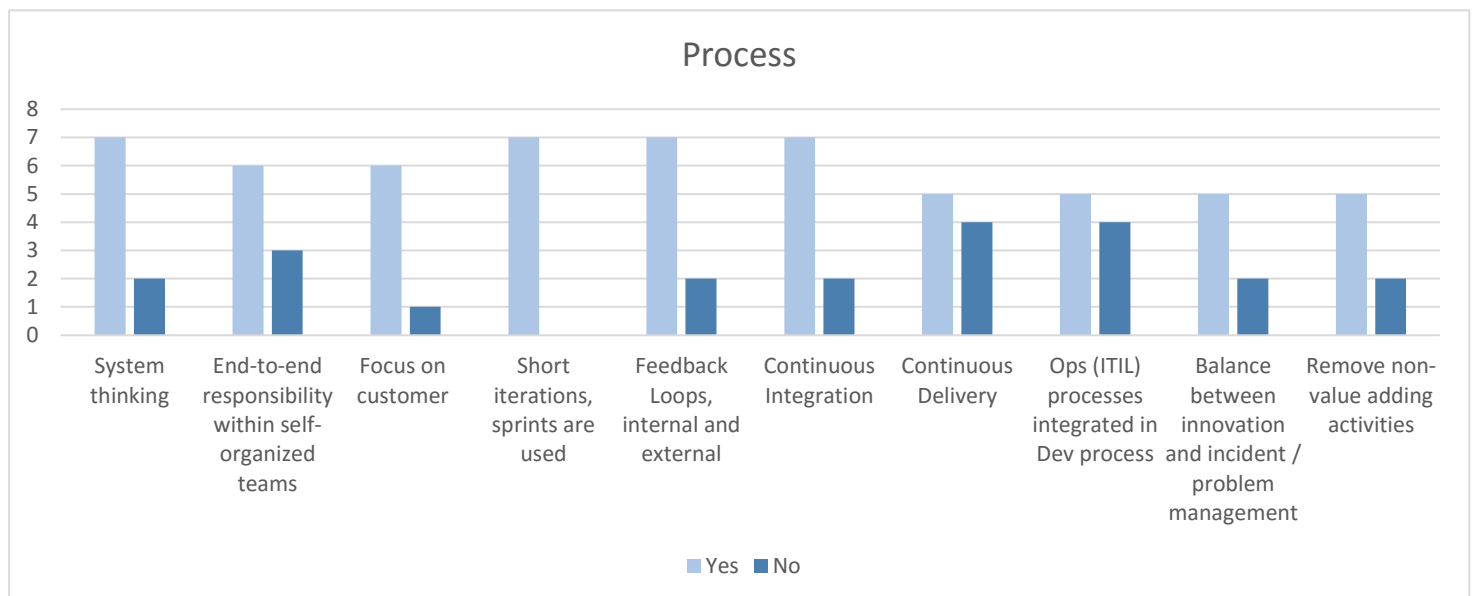


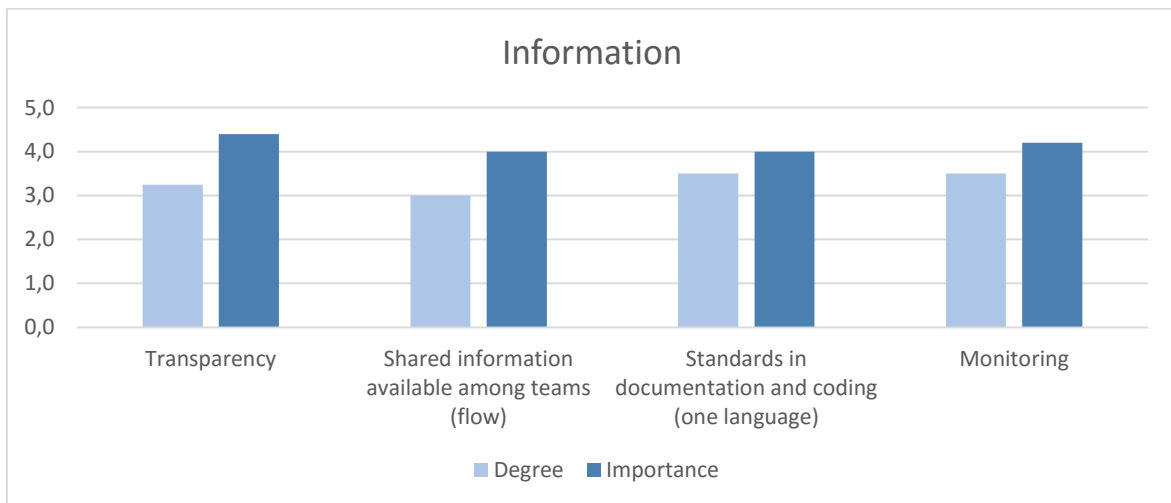
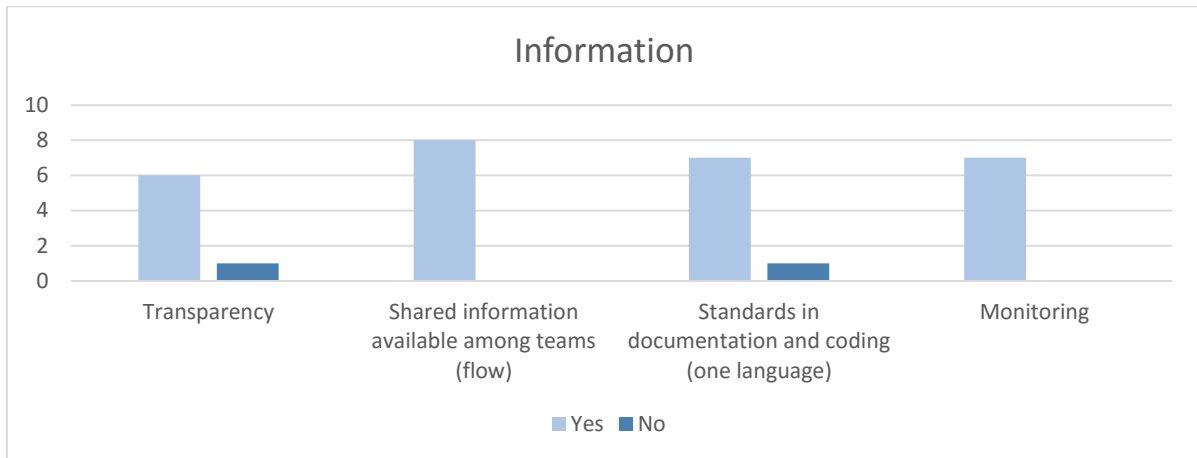
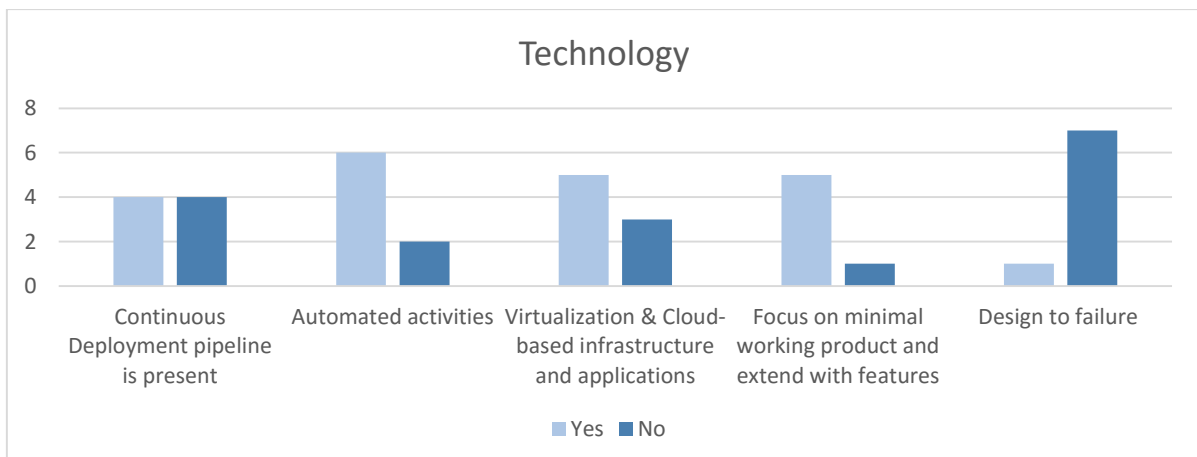
Org. Culture & Structure

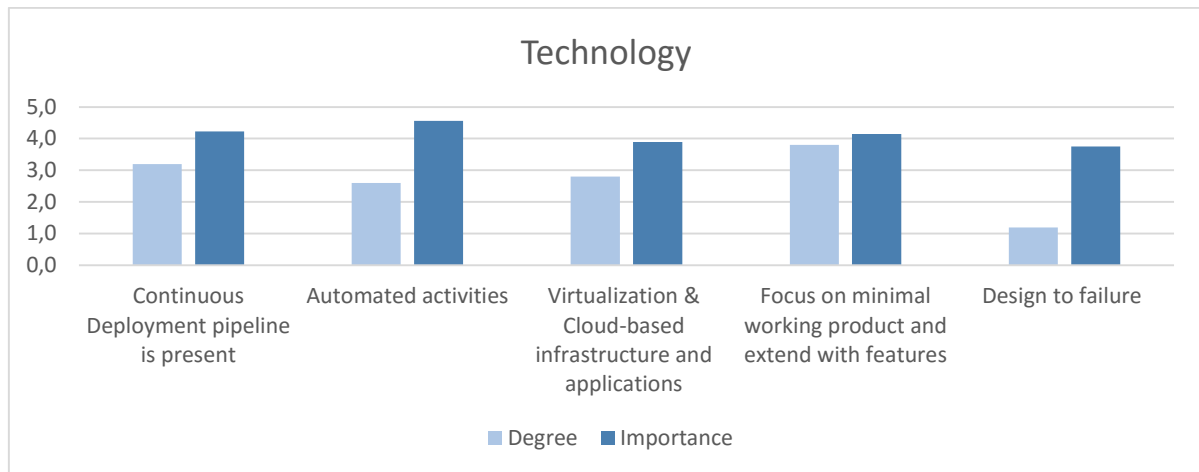




Process



Information**Technology**

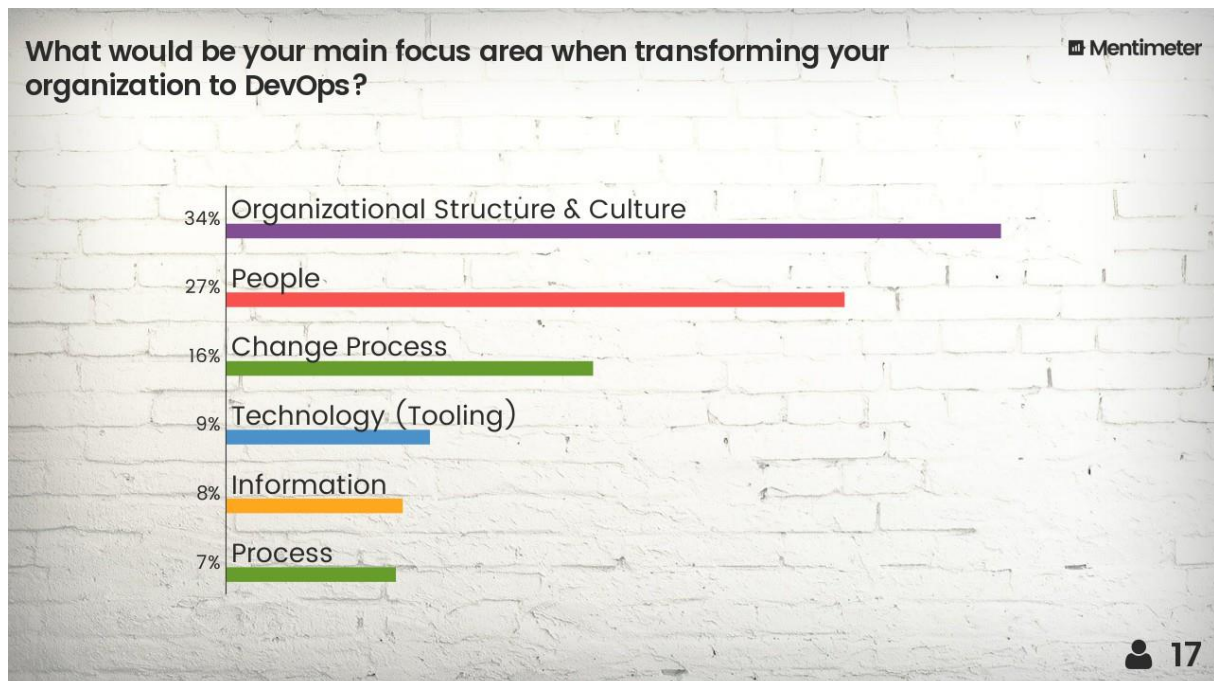


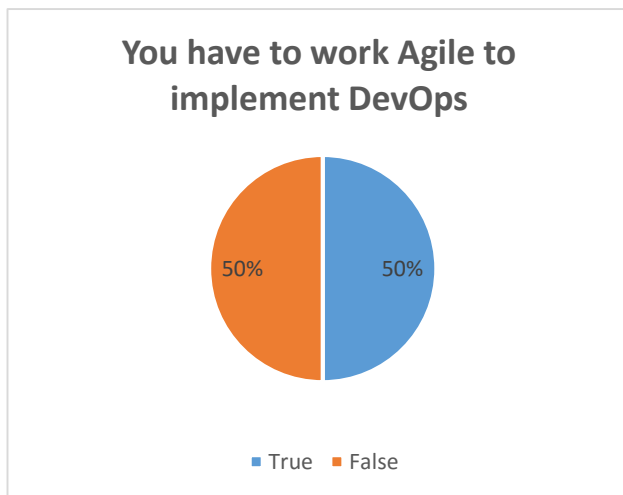
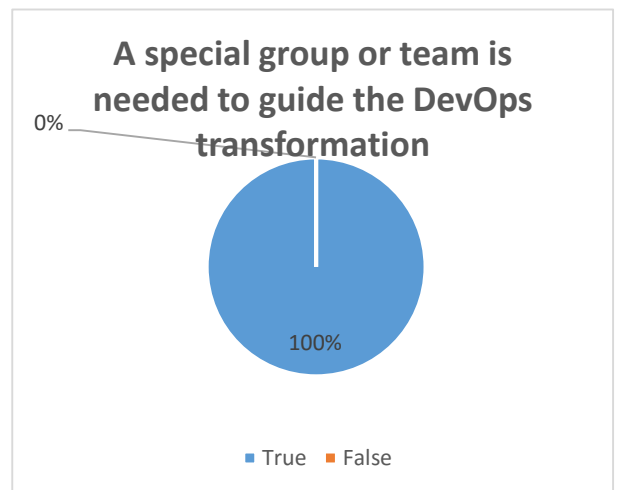
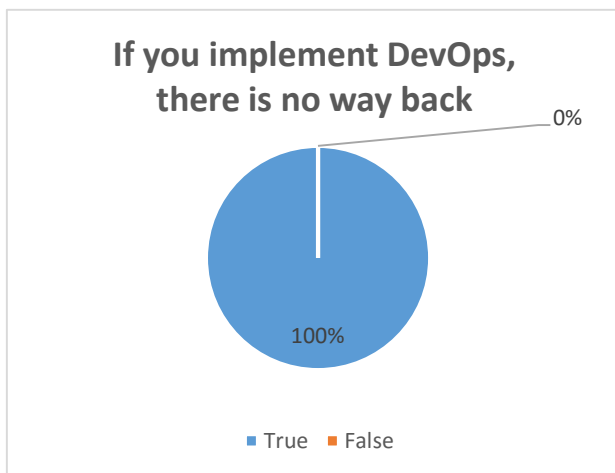
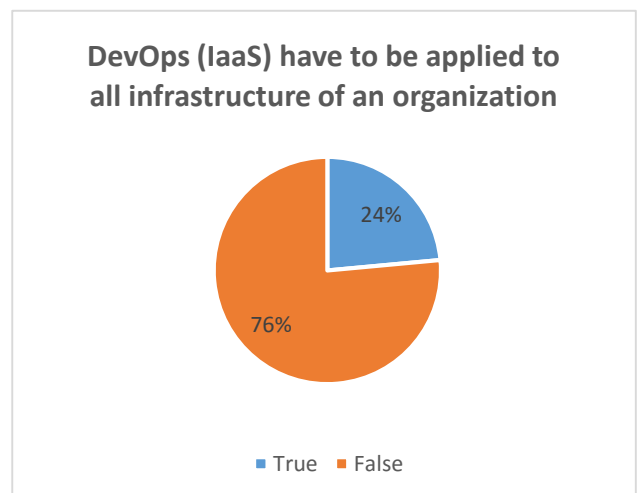
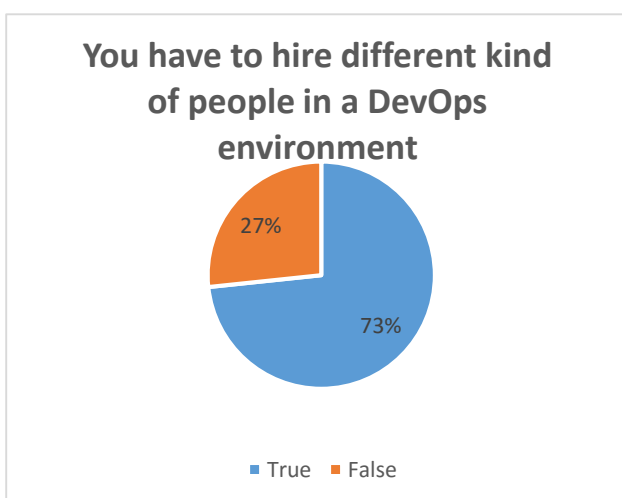
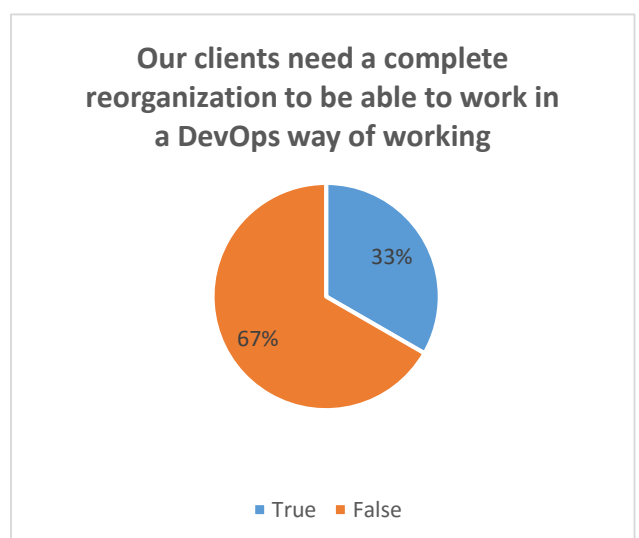
Appendix G – Results Evaluation Session

Question 1 (Participants: 16)



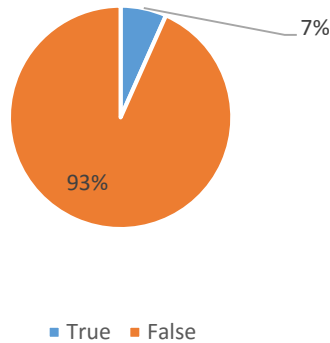
Question 2 (Participants: 17)



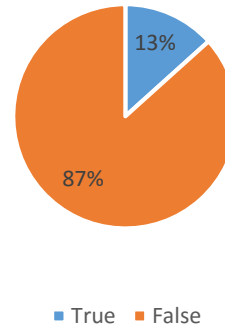
Question 3 (Participants: 16)**Question 4 (Participants: 13)****Question 5 (Participants: 14)****Question 6 (Participants: 17)****Question 7 (Participants: 15)****Question 8 (Participants: 15)**

Question 9 (Participants: 15)

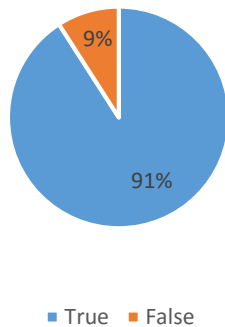
Large organizations have too many existing contracts and legacy to adopt DevOps

**Question 10 (Participants: 15)**

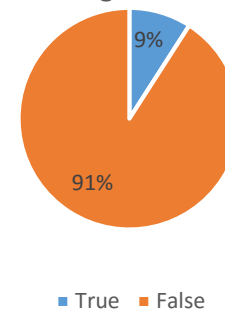
A DevOps transformation should be approached from bottom-up, in the following order:
1. Individual, 2. Team, 3. Organization

**Question 11 (Participants: 11)**

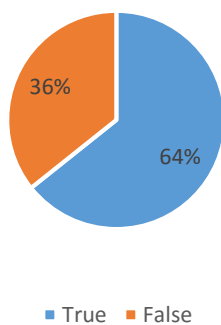
DevOps is not implemented in clear phases, but in a continuous experimental process

**Question 12 (Participants: 11)**

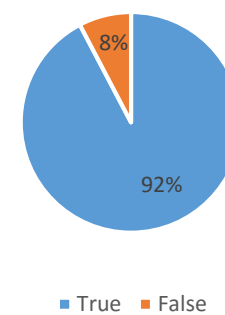
Organizations have to adopt to a FULL DevOps way of working. There is no such thing as half DevOps

**Question 13 (Participants: 14)**

The physical environment has to be adjusted in order to encourage a DevOps way of working

**Question 14 (Participants: 13)**

Different compliancy rules and laws are limiting organizations from adopting a DevOps way of working



Appendix H – Implementation Assessment Tool

What would be/are the main reasons to adopt a DevOps way of working?			(1 = low, 5 = high) or Low/Medium/High					(1 = low, 5 = high) or Low/Medium/High				
External Customer and Supplier Power			1	2	3	4	5	1	2	3	4	5
Industry Competitiveness			1	2	3	4	5	1	2	3	4	5
When implementing DevOps, which aspects would be/were the most important to focus on?	Indicate whether these aspects are already applied in the current situation:		Indicate the degree of each aspect at your organization:					Indicate the degree of each aspect at your organization:				
(Business) Process												
Continuous Integration	Yes	No	1	2	3	4	5	1	2	3	4	5
Continuous Delivery	Yes	No	1	2	3	4	5	1	2	3	4	5
System Thinking	Yes	No	1	2	3	4	5	1	2	3	4	5
End-to-end responsibility in self-organizing team	Yes	No	1	2	3	4	5	1	2	3	4	5
ITIL processes in Dev Processes	Yes	No	1	2	3	4	5	1	2	3	4	5
Balance between dev, innovation and problems	Yes	No	1	2	3	4	5	1	2	3	4	5
Remove non-value adding activities	Yes	No	1	2	3	4	5	1	2	3	4	5
ITIL processes in Dev Processes	Yes	No	1	2	3	4	5	1	2	3	4	5
Feedback loops	Yes	No	1	2	3	4	5	1	2	3	4	5
Focus on customer	Yes	No	1	2	3	4	5	1	2	3	4	5
People												
Experimenting without regret	Yes	No	1	2	3	4	5	1	2	3	4	5
Learning culture	Yes	No	1	2	3	4	5	1	2	3	4	5
Cross-skilled people	Yes	No	1	2	3	4	5	1	2	3	4	5
Willingness	Yes	No	1	2	3	4	5	1	2	3	4	5
Sharing knowledge	Yes	No	1	2	3	4	5	1	2	3	4	5
Different type of people/skills set	Yes	No	1	2	3	4	5	1	2	3	4	5
Proactive mind set and attitude	Yes	No	1	2	3	4	5	1	2	3	4	5
Understand the reason to change	Yes	No	1	2	3	4	5	1	2	3	4	5
Cross-functional teams	Yes	No	1	2	3	4	5	1	2	3	4	5
Technology												
Design for Failure	Yes	No	1	2	3	4	5	1	2	3	4	5
Automation	Yes	No	1	2	3	4	5	1	2	3	4	5
Virtualization	Yes	No	1	2	3	4	5	1	2	3	4	5
Build new technology next to monolithic system	Yes	No	1	2	3	4	5	1	2	3	4	5
Focus on minimal valuable product	Yes	No	1	2	3	4	5	1	2	3	4	5
Continuous Deployment	Yes	No	1	2	3	4	5	1	2	3	4	5
Information												
Transparency	Yes	No	1	2	3	4	5	1	2	3	4	5
Shared information	Yes	No	1	2	3	4	5	1	2	3	4	5
Monitoring	Yes	No	1	2	3	4	5	1	2	3	4	5
Standards	Yes	No	1	2	3	4	5	1	2	3	4	5
Organizational Culture & Structures												
KPIs	Yes	No	1	2	3	4	5	1	2	3	4	5
Incentives	Yes	No	1	2	3	4	5	1	2	3	4	5
Top management support	Yes	No	1	2	3	4	5	1	2	3	4	5
Culture of collaboration	Yes	No	1	2	3	4	5	1	2	3	4	5
Change in organizational structure	Yes	No	1	2	3	4	5	1	2	3	4	5
Open and creative physical environment	Yes	No	1	2	3	4	5	1	2	3	4	5
Performance measurement												
Value	Yes	No	1	2	3	4	5	1	2	3	4	5
Quality	Yes	No	1	2	3	4	5	1	2	3	4	5
Time	Yes	No	1	2	3	4	5	1	2	3	4	5
Cost	Yes	No	1	2	3	4	5	1	2	3	4	5
Capacity/velocity of the team	Yes	No	1	2	3	4	5	1	2	3	4	5
Customer Satisfaction	Yes	No	1	2	3	4	5	1	2	3	4	5
Flexibility/Innovation	Yes	No	1	2	3	4	5	1	2	3	4	5

Appendix I – Scientific Article

Critical Aspects for the Implementation of DevOps at Large IT Service Organizations

Margot Jonker (4397975)

Faculty of Technology, Policy and Management

Delft University of Technology

m.a.j.jonker@student.tudelft.nl

Abstract

Currently the business world is focused more and more on increasing flexibility within their processes and decreasing the time-to-market. With this purpose, iterative and Agile approaches conquer ground within business processes. A relatively new approach within the software development process is working according to DevOps, which could be interpreted as applying an extension of Agile practices to a broader field within an organization that involves IT services. Large IT service organizations seem to struggle with the implementation of DevOps as they have all kind of legacy that make it hard to transform their business processes. Therefore, it can be questioned on which aspects of DevOps they have to focus. This paper aims to identify the critical aspects on which a large IT service organization has focus their DevOps implementation. A literature research is performed to define DevOps and its practices, where after a business process change approach is chosen to identify which elements of the business process are impacted by a DevOps way of working. Based on nine case interviews and an expert evaluation session, the relevant aspects of DevOps for large IT service organizations are identified. Although it is found that it is crucial to focus on people and change management, the business process components are dynamic in relation to DevOps. Besides, the implementation of DevOps aspects is highly dependent on multiple organizational factors. The paper suggests future research in the areas of quantitative research on the focus areas, a comparison with small organizations and a systems dynamic approach on DevOps.

Key words: *Agile, DevOps, business process change, IT services, critical aspects, implementation*

1. INTRODUCTION

Technological innovations make the world of today more and more digitalized. Being flexible and being able to innovate at a fast pace is key nowadays for organizations to stay competitive (Salehi & Yaghtin, 2015). Many organizations within different sectors are thereby changing their business strategy, often resulting in offering or implementing Information Technology (IT) services. These IT services can be the new main business of an organization or a supportive service of the main business. IT service organizations try to adapt to this rapidly changing environment and look for ways to make their business process more responsive to change. Because of this, iterative and Agile approaches have conquered ground within the business processes of IT service organizations. Agile approaches focus on aligning the development phase better with the business requirements, but leave the operational phase out of scope. In contrast, to align the whole

business process, a relatively new trend, called DevOps, emerged. DevOps focuses on bringing the development and operational department together and can be seen as an extension of the Agile scope to increase the speed of the delivery process. DevOps is a portmanteau of development and operations and seems to apply to the delivery lifecycle of IT services (De Bayser et al., 2015). However, there is no consensus on the definition of DevOps. This paper uses the following definition that is based on a review of multiple definitions found in literature (Bass, Weber, & Zhu, 2015; Colavita, 2016; De Bayser et al., 2015; Erich et al., 2014; Hüttermann, 2012; Kim, 2011; Walls, 2013):

“DevOps is a cultural movement that breaks silos between the business, development and operations department, combined with a number of service development practices regarding people, process and technology, that enable rapid development and delivery of IT services”

As DevOps promises to enable a rapid development and delivery of IT services, IT service organizations desire to implement this new trend within their business process. Therefore, the implementation of DevOps at large IT service is the object of this study. Such organizations are often struggling on which aspects they have to focus the adoption of DevOps. In order to identify the focus aspects, the following research question is formulated:

On which aspects does a large IT service organization have to focus its implementation of a DevOps way of working?

This paper aims to identify the critical aspects on which a large IT service organization has to focus its DevOps implementation. In order to identify these aspects, a research is conducted with using literature research, case interviews and an expert evaluation session. Literature research is used to identify the general areas and aspects for DevOps. Based on this general approach of DevOps, case interviews at different large IT service organizations are used to identify the most relevant aspects specifically for such organizations. The expert evaluation session is used to evaluate the additional findings of the case interviews.

The paper starts in section 2 with describing DevOps and its values. This leads to the identification of the challenges related to the implementation of DevOps. The empirical research that is conducted for this paper is described in section 3. Section 4 discusses the empirical results that are found during the cases and the evaluation session. The answer to the research question is given in section 5 as conclusion of the paper. This section is followed by sections 6 and 7 that reflect on the research and give recommendations for further research.

2. DEVOPS

Academic literature on DevOps is rarely available, neither on the implementation of DevOps. The literature that is available approaches DevOps often from a certain perspective and describe the practices that are included. For example, Hüttermann (2012) takes a developer approach to DevOps, Bass et al., (2015) perceives DevOps from the viewpoint of the software architect and others are combining it with Continuous Delivery (Humble & Farley, 2010; Swartout, 2014). De

Bayser et al. (2015) explore the usefulness of DevOps concepts to improve the development of software that supports scientific projects. They present some best practices to reduce friction in software development, such as “continuously integrate changes to foster discussion and fast validation of peers” (p. 1403). Walls (2013) focusses on building the DevOps culture.

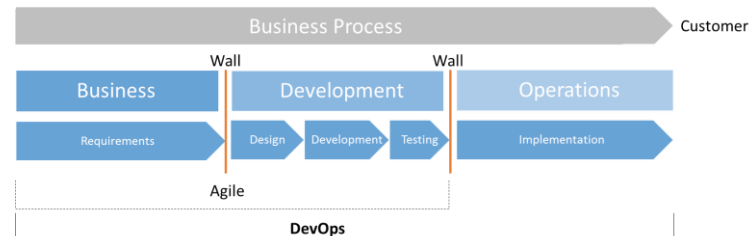


Figure 51: Overview of the software delivery pipeline with Agile & DevOps indicated

Within this paper, DevOps is seen as an extension of Agile methodology, as can be seen in Figure 51. Within DevOps, operations will become a valued member of the traditional Agile process with equal rights (Debois, 2011). To extend the Agile way of working to the entire business process, DevOps involves also elements of other methodologies, such as Lean Thinking and ITSM. The differences between a traditional, an Agile and a DevOps way of working as defined within this research are shown in Table 19.

Values

Based on the Three Ways of Kim (2013) and the additional fourth way of Edwards (Pais, 2012), the values of DevOps are in this research defined as: systems thinking, focus on flow, amplify feedback loops and culture of continual experimentation and learning (Figure 50).

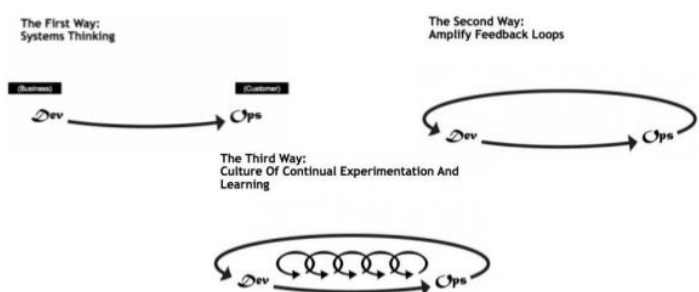


Figure 50: The three ways of DevOps (Kim et al., 2013)

Systems thinking refers to the focus is on all business value streams that are enabled by IT, from the business or IT requirements to the end

where the value is delivered to the customer as a service (Kim, 2013). The focus on flow relates to this way of thinking as it is important to ensure the flow of the complete process from the beginning to the end and not only focusing on one phase or silo. To be able to create this flow and to respond to internal and external customers quicker, amplifying feedback loops throughout the whole process is crucial, which relates to the Agile way of working. As last, Kim (2013) states the importance of creating a culture that embraces continual experimentation, taking risks and learning from failure on the one hand and understanding that repetition and practice is the prerequisite to mastery on the other hand.

Challenges

As Iden et al. (2011) state, the importance of IT operations throughout the whole process must be acknowledged. This is not limited to the software delivery process, the DevOps way of working can also revitalize other business processes (Swartout, 2014). Hence, adopting a DevOps way of working in an organization can be approached as a business process change (Swartout, 2014). Based on different theoretical models for business process change (Jurisch et al., 2012; Kettinger & Grover, 1995; Mayer & Benjamin, 1995), we defined general focus

areas of the business process for the change towards a DevOps way of working.

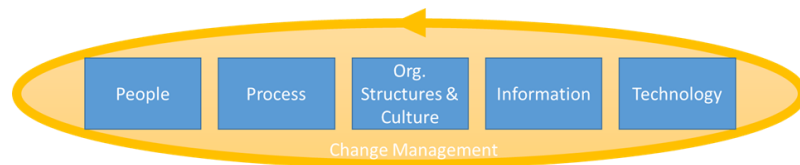


Figure 52: Components for implementation of DevOps

Regarding a DevOps implementation, the areas in which organizations face different challenges are people, process and technology. These areas match the holistic model for process improvement (Prodan et al., 2015). Next to that, the way in which information is shared and available across the whole business process is crucial for DevOps (Prodan et al., 2015; Van Bon, 2004). Also, the organizational culture and structure is impacted by DevOps and relates to the environment in which process, people and technology interact (Jurisch et al., 2012). Management of all changes should be taken into account as the implementation of a new way of working will never have a chance to be successful without managing the changes carefully (Grover et al., 1995). Organizational change management is defined by Moran & Brightman (2000) as “the process of continually renewing an organization’s direction, structure, and capabilities to serve the ever-changing needs of external and internal customers” (p. 111).

Table 19: Overview of differences between waterfall, Agile and DevOps

Category	Traditional Adapted from Source: (Nerur et al., 2005)	Agile Adapted from Source: (Nerur et al., 2005)	DevOps within this research
Fundamental Assumptions	<i>Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning</i>	<i>High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change</i>	High-quality, adaptive services can be delivered per multidisciplinary end-to-end teams using the principles of continuous improvement, integration, delivery and possible deployment
Role Assignment	<i>Individual – favors specialization</i>	<i>Self-organizing teams – encourages role interchangeability</i>	Multidisciplinary teams with end-to-end responsibility
Communication	<i>Formal</i>	<i>Informal</i>	Informal
Customer’s Role	<i>Important</i>	<i>Critical</i>	Critical
Project Cycle	<i>Guided by tasks or activities</i>	<i>Guided by product features</i>	Guided by delivering business value from end-to-end
Development Model	<i>Life cycle model (waterfall, Spiral, or some variation)</i>	<i>The evolutionary-delivery model</i>	Continuous Delivery Model
Desired Organizational Form/Structure	<i>Mechanistic (bureaucratic with high formalization)</i>	<i>Organic (flexible and participative encouraging cooperative social action)</i>	Learning (Interactive, flexible and collaborative people and teams)
Technology	<i>No restriction</i>	<i>Favors object-oriented technology</i>	Automation (Cloud, XaaS)

Therefore, by taking a business process change approach, the research performed for this paper focuses on six focus areas, hereafter called components: people, process, technology, information, organizational culture & structure and change management (Figure 52). Aspects of DevOps within these components will be used to identify the most relevant focus points for large IT service organizations, from the perspective of an IT manager (Table 21).

3. EMPIRICAL RESEARCH

Nine cases within eight different IT service organizations are involved within our research to identify the most relevant aspects of the implementation of DevOps at large IT service organizations. An overview of the cases and the different interviewees can be found in Table 20. Semi-structured interviews with IT managers or similar functions have been conducted with using a theoretical interview guide. The criteria for selecting the organization and interviewee(s) per case were based on providing an IT service, organization's size, current way of working, DevOps intention and availability. The requirement for the current way of working was that the Agile way of working had to be implemented already as DevOps is seen as an extension of Agile within our research.

Table 20: Overview of the cases

#	Operating Group	Interviewee(s)
1	Financial Services	IT manager/ Chapter lead
2	Communications & High Tech	IT manager /Agile coach + Chain manager
3	Communications & High Tech	Infrastructure consultant
4	Products	IT manager Ops. + IT manager Dev.
5	Products	IT manager / Product owner
6	Financial Services	IT manager
7	Financial Services	2 Tech. architects
8	Communications & High Tech	Service Manager
9	Products	Development Manager

Some additional findings of the cases have been assessed by experts during an evaluation session. Several statements were formulated on basis of the findings and discussed by the experts. An online tool was used to gather everyone's input and extra comments were noted by the researcher.

During the case interviews, the interviewees were asked to indicate the degree of implementation of each aspect within their department or team. Also the importance per aspect from their point of view was questioned. By comparing the outcomes of the interviewees, the gaps are indicated and associated comments were analyzed, which resulted in a ranking of the most relevant focus areas. Different categories were also defined as: aspects that were already implemented due to an Agile way of working, aspects that are relevant for DevOps and aspects that would be nice to have.

4. FINDINGS

In the next sections the focus areas are discussed and the associated aspects are discussed per component. As last, the organizational factors are discussed. The overall results are shown in Table 21. Within the table some aspects are abbreviated.

Varying focus

Our empirical data shows that all six components are relevant for the implementation of DevOps within a large IT service organization. They are interconnected and change impacts all of them in certain degree. All components are important to take into account with a DevOps implementation, although they can vary in focus. The component people has a priority to focus on, as well as the overall change management component. Within our research, we defined DevOps as a cultural movement and the importance of the cultural aspect is confirmed by our empirical data. The interviewees state that the implementation of DevOps will not reach its successfulness when people are not taken into account in the beginning. The components information and technology are facilitators while those components can also be crucial during the implementation.

We will discuss the identified aspects for each component. The order in which we discuss the aspects relates to the common importance as

indicated by the interviewees, starting with the most important aspects on top.

Focus on Change Management

Within the change management component, most interviewees stated that all of the identified aspects are important, however, there is still a certain ranking in which aspects are perceived relevant. As the importance of IT operations should be acknowledged throughout the whole process (Iden et al., 2011), it is important to treat employees of the development and operations departments equally. Conflicting interests can be solved when both departments understand each other. Feedback sessions on the process are included in the Agile way of working (Boehm & Turner, 2005), but our empirical data shows that this could be improved as feedback is still more focused on the performance of the service than on the way of working.

Next to that, understanding of the DevOps way of working is indicated as crucial. Most interviewees state this is already present at employees within the team itself, however, all

employees should understand it as it impacts the whole business process. Continuous improvement is also indicated as important as one manager states: *“It is never finished”*. The aspect that could help to improve this is to show results and make improvements on the process visible. This creates more awareness of the change efforts of people. Additionally, training within the Agile way of working is often already in place. However, it is argued that training of the DevOps way of working should be done with working within a DevOps environment, as learning on the job is important to understand it. It is agreed by most interviewees that a special group to support the transformation to DevOps is useful, however, often Agile coaches are



Table 21: Overview of focus areas and their aspects

Figure 53: Legend associated to Table 21

	Process	People	Information	Org. Culture & Structure	Technology	Change management
Start Focus	2	1	3	2	3	1
Relevant aspects	Continuous Integration	Experimenting without regret	Transparency of org. information	KPIs	Automation	Treat employees equally
	Continuous Delivery	Encourage learning culture	Shared information	Incentives	Virtualization	Feedback sessions
	Systems Thinking	Willingness	Monitoring	Change in organizational structure	Build new technology next to monolithic system and replace	Create understanding
	End-to-end responsibility within self-organizing teams	Cross-skilled people	Standards for coding	Top management support & commitment	Design for Failure	Continuous improvement
	Focus on value	Sharing knowledge		Culture of collaboration	Continuous Deployment	Show results
	ITIL processes in Dev processes	Proactive mindset and attitude at the people		Budget, time and room to move towards new way of working	Focus on MVP	Training
	Balance between dev, innovation and problems	Different type of persons, based on its skills set which should include a basic and broad understanding of multiple areas		Environmental factors have to be adjusted as well, create a more creative and open place		Assign group
	Conflicting processes have to be aligned within the DevOps teams	Commitment of people involved		The KPI 'value' has to be translated in different metrics at different stages		Make an overview of automation
	Sprints	Cross- functional teams				People understand the reason of the change
	Feedback loops					

already involved in the process, which reduces the need for a separate support group. Making an overview of the automated activities is indicated as less important as the focus should not be on documentation, but on people and the activities change too often to make an elaborative overview.

As an additional finding, the aspect of understanding the reason of the change is mentioned. Large IT service organizations have very often many employees that work already for a longer period at the organization and are used to the way they are currently working. In order to take them along in changes, they have to understand the reason to change. This is also agreed by the experts in the session.

Focus on People

The focus on people is indicated as highly important. First of all, experimenting without regret is indicated as important, but is not yet present at all. It refers to a blameless culture, where employees are not completely free to do whatever they want, but are able to experiment within their tasks in order to learn and operate innovatively (Davis & Daniels, 2015). A learning culture is related to this blameless culture (Swartout, 2014) and is also not yet adopted in a high degree. Next to that, the willingness of people (Colavita, 2016) is already present at a certain degree most of the time, however, it should be maintained as it is very important that people keep going along with the DevOps culture. Cross-skilled people (Edwards, 2014) are less present at the case organizations yet, but are indicated as important in order to create a basic understanding of all tasks among all employees. However, it has to be acknowledged that people will remain expert in their own expertise. The degree in which knowledge is shared within the organizations is already at a certain level. It should only be ensured that the knowledge is kept in the organization and not only within the people.

As additional findings within this component, a proactive mindset and attitude is needed at the people involved, as well as commitment of everybody, although this is not specially evaluated through the expert evaluation session. Next to that, the interviewees mentioned that the requirement for a basic and broad understanding of multiple skills different types of people should be employed, however, after the expert

evaluation this is seen as a nice to have. It was stated that the actual hiring of completely new people does not happen within large organizations, although DevOps skills can be taken into account in the HR model for hiring new roles if desired.

According to the Agile way of working, cross-functional teams are already formed. Debois (2011) indicates that many organizations do create small teams, however, they make the mistake of designing them functionally based on technology and not on a service or feature. Our empirical data shows that most organization have already, although informal, cross-functional teams, but the operational function is not yet completely included. This should be improved, however, it is not always possible for these organizations to integrate the operational department completely into these teams due to organizational factors.

Focus on Process

Focusing on the process, continuous integration can be improved at most organizations, mostly regarding the technological part of automation. Continuous integration and continuous delivery are highly dependent on automation and therefore, connected to the technology component. Continuous integration means that there are no major updates to the system configuration, but updates are only possible in series of small changes in the system (Schaefer, Reichenbach, & Fey, 2013). As continuous delivery is the next step after continuous integration, this can also be improved, but logically the focus on continuous integration has to be first. Humble & Farley (2010) state that “continuous delivery is a way of working whereby quality products, normally software assets, can be built, tested and shipped in quick succession—thus delivering value much sooner than traditional approaches” (Abstract). In relation with the values of DevOps, systems thinking has to be important throughout the whole process (Kim, 2013) and this can be improved within the organizations. This approach results in the emergence of end-to-end responsibility within self-organized teams, where self-organizing teams are already described within Agile Manifesto (Beck, 2001). Within Agile, the focus has to be on the customer, where in DevOps the focus has to be on delivering business value. Our empirical data shows that the focus on the customer is already

implemented, however, focusing only on adding value can be improved. Most managers state it is less important to transform ITIL process into DevOps processes, however, they do state that process alignment is important and ITIL process have to be adjusted.

As an additional finding our empirical data shows the importance of aligning the conflicting interests of both departments. Operations is often used to a 24/7 culture where people are standby at all moments, where development is more designed according to a project and functional basis. This relates mainly to the processes that are used. The development department is responsible for creating new business value and the operational department is focusing on minimizing disruptions of the system. Next to that, our empirical data shows that sprints and feedback loops are already implemented through the Agile way of working. Organizations do not have to focus on this anymore, only ensure to maintain these aspects in the way of working. Most managers state that removing non-value adding activities is important throughout the whole business and not particular or DevOps. They do mention that the mindset of continuous improvement at people could be improved, but other aspects have more influence on this.

Focus on Organizational Culture & Structure

As a main focus in this component, the KPIs have to be adjusted. The main performance indicator will become business value and the impact of a certain team or function on this value. The KPI “value” has to be translated in different metrics at different stages. Many interviewees acknowledge the importance of value as KPI, however, they still struggle how to translate this into real metrics. Next to that, the incentives of everybody involved in delivering software have to be aligned (Debois, 2011). Therefore, there are not individual incentives anymore on the delivering value, but only incentives on the team effort on a particular business stream or value (Hüttermann, 2012). Most managers acknowledge again that measurements should be based more on team performance in order to create common objectives. Changing the organizational structure is varying per organization. Different organizational models that embrace an Agile or DevOps way of working are used within the cases and other organizations have only

changed some roles for an Agile way of working. This is often only on a lower level, at team basis. The entire structure of an organization is often not changed, as this is seen as impossible due to multiple organizational factors, although it could improve a DevOps way of working. Associated to this, our empirical data shows that top management supporting and facilitating the process is already implemented to a certain degree, however, it can be improved at almost all organizations. This aspect is also influencing other aspects and managers state that the support and understanding of DevOps can be improved at certain moments during their work. As last, the culture of collaboration is already implemented, however, this can always be improved as the degree of collaboration within DevOps is higher.

An additional finding within this category, is that there has to be budget, time and room to move towards a new way of working. Next to that, our empirical data shows that the environmental factors have to be adjusted as well in order to create a more creative and open place. An example of this could be flexible work spaces or removing physical walls. However, during the expert evaluation some experts stated that a traditional working environment with strict offices would not prevent a successful DevOps implementation. Based on this, it can be seen as a nice-to-have, as it is not mandatory nor irrelevant and it can positively affect the implementation of DevOps.

Focus on Information

In order to create a focus on flow, transparency of organizational information involved is necessary. An environment is necessary in which anyone and everyone feels that they can speak their minds, and more importantly, contribute, as Swartout (2014) mentions. From our empirical data, transparency of information seems to be hard with working in silos. Managers identify that within a team information is being shared, but on the organizational level it is still limited. To increase the flow within the process, all necessary information on the process and product have to be shared across all DevOps teams (Hüttermann, 2012). Lack of transparency between dev and operations is appointed as a problem, as these teams are not

working together yet within Agile. Effective monitoring of information also supports the collaboration between the departments (Lwakatare, Kuvaja, & Oivo, 2015). This information can also be analyzed and used as feedback to improve the entire service or process. Monitoring on the service performance is at some organizations being done by large dashboards or screens in the working spaces, although monitoring of the team performance could be improved according to the interviewees.

Erich et al. (2014) states that “developers and operations should try to make their documentation understandable by both sides” (p. 13). This can be achieved by agreeing upon standards in documentation and coding, which is often already done within the Agile way of working at the case organizations.

Focus on Technology

To reach for continuous delivery, automation of activities is key. Automation offers possibilities to make tasks more efficient (Erich et al., 2014). Most organizations have already undertaken certain steps towards automation and are still busy with extending this, due to the fact that this is already important within the Agile way of working. Next to that, virtualization & cloud-based infrastructure and applications is also implemented to a certain extent already within the organizations, although this is less applicable for the operational team and for some infrastructure departments. Due to outsourced infrastructure or legacy systems, it is not always possible or beneficial to transform all systems into software based systems.

The additional finding within the technology component is that it was stated that some organizations build new technology parallel to their former monolithic systems. This is often done in order to be able to deliver on-going services during the change management and by doing so, customers notice the least from the change towards a new system or technology. The new system is build next to the former one and once the new system is mature enough, they remove the old system. However, it is often seen that still some infrastructure parts of the old system keep on existing due to the absence of a beneficial business case to replace them.

Design for failure and continuous deployment are nice-to-have's in this category. Design for

failure means not preventing failure, but designing resilient services that can survive failures and ensures its availability and reliability (Abbadi, 2011). This is highly related to Cloud applications and these are not yet implemented at a large scale within the large organizations. Other technology aspects are more important to focus on before thinking about this. Continuous Deployment is the next step after continuous delivery and can be defined as deploying every change automatically to production whenever it is ready (Ten Hagen & Heunks, 2016). Continuous deployment is not desired as main goal for each organization. It is possible that changes to the system have to be approved by certain people as certain laws or rules that are imposed to an organization (Ten Hagen & Heunks, 2016). Therefore, continuous deployment should only be the goal of organizations which are not constrained by regulatory (Caum, 2013). Finally, the focus on the minimal valuable product is already adopted through the Agile way of working.

Organizational factors

Although our empirical data shows a certain common ranking of the aspects by the interviewees across the IT service organizations, it also becomes clear that a DevOps implementation is a complex and dynamic process that differs per organization. The implementation of DevOps is highly dependent on the organizational context and its characteristics.

First of all, the physical environment seems to have an impact on the success of DevOps. Working in an open and creative workspace enables people to work together. Next to that, the overall organizational context influences the implementation of DevOps. Organizational factors such as outsourcing of functionalities or even whole departments, compliancy rules and laws, existing contracts, organization's overall culture and a complex landscape with legacy in technology and stakeholders make it necessary for an organization to adapt its DevOps implementation to specific organizational characteristics. Therefore, it is not possible to appoint a standardized way that every organization can follow for the DevOps implementation. This is in line with the contingency theory that states that organizational characteristics have to fit their

contingencies (environment, organizational size and strategy) in order to lead to high performance (Donaldson, 2006). Therefore, the way in which DevOps is implemented has to match with the organizational factors. Next to that, DevOps should only be implemented in those areas where it will likely deliver the most value, which differs per type of service, per department and per specific organization.

5. CONCLUSION

In this paper we presented our research project into the implementation of DevOps at large IT service organizations. The objective of this paper is to answer the following question: *“On which aspects does a large IT service organization have to focus its implementation of a DevOps way of working?”*. Insights from literature research were used to identify the relevant areas and associated aspects for DevOps. Case interviews at large IT service organizations were used to identify the most relevant aspects to focus on when implementing DevOps at such organizations.

The paper identifies the categories that are impacted by DevOps as: people, process, technology, information, organizational culture & structure and change management. All components are interconnected and important to take into account within a DevOps implementation at a large IT service organization, although the people and change management are identified as most important to focus on at the start of the implementation of a DevOps way of working.

Each category includes certain aspects that relate to a DevOps way of working. Based on our empirical data, the importance is formulated for large IT service organizations to focus on (Table 21). This ranking is formulated based on the empirical data of the nine cases and could be seen as the common order of aspects for a large IT organization that applies an Agile way of working and wants to move towards a DevOps way of working.

However, the implementation of DevOps is highly dependent on organizational factors. The organizational factors are identified as physical environment, outsourcing of functionalities or even whole departments, compliancy rules and laws, existing contracts, organization’s overall culture and a complex landscape with legacy in

technology and stakeholders. Also the type of service, department and organization influence the way DevOps is implemented. This means that there is not a standardized way to follow for all organizations. The ranking should not be seen as a fixed order and organizations should focus on the aspects that are most relevant for them, based on their current way of working and their organizational factors.

6. REFLECTION

DevOps is a relatively new term, which means that there is not much scientific literature available yet. There is no common definition and also the practices that are included in DevOps are not commonly shared. The research uses a research specific definition. Therefore, it is not possible to compare it with possible future research that defines DevOps differently. Also the practices that are used within the research are specifically formulated, based on available literature as much as possible. The placement of the aspects within the categories is done by the researcher, which implies subjectivity. Other researchers could categorize the aspects differently.

Next to that, the research makes use of case interviews with interviewing one or two IT managers or similar roles per case. As the degree of current implementation and importance is asked only to them within the organization, the research is influenced with subjectivity of the interviewees. The semi-structured interviews make use of a small quantitative indication of the implementation degree and importance, however, this cannot be significantly proven as the data set is too small. Therefore, these outcomes of this research cannot be generalized for all large IT service organizations and should be seen as an exploratory, mainly qualitative research project on the focus areas for DevOps implementations within large IT service organizations.

Whereas not all additional aspects were evaluated in the evaluation session, the additional aspects were mentioned multiple times, which makes them likely to be relevant for DevOps.

7. FUTURE RESEARCH

Based on the conclusions and reflections, we formulate the following suggestions for future research:

- Within the research, empirical research is done to discover the focus areas for a DevOps implementation. Future research should aim at quantitative research, for example a survey on the implementation of these areas. On the other hand, a more in-depth case study at one organization could give insights into different perspectives on DevOps within one organization.
- This research was focused on large IT service organizations. As it seems to be easier to implement DevOps at smaller organizations, it could be interesting to compare the ways of implementation between smaller and larger organizations.
- As the environment in which DevOps operates can be seen as complex, the implementation of DevOps can be approached from the theory of complex adaptive systems. The results of this research are highly dependent on the choice of a business process change approach. Taking another approach than a business process change approach can lead to other aspects of DevOps implementation that are relevant for the transformation process, for example using the Delphi method to analyze the judgement of multiple experts to create a consensus on the DevOps definition and practices. Next to that, a systems dynamic approach could analyze the influence of all aspects involved in the complex environment.

REFERENCES

- Abbadi, I. (2011). Self-managed services conceptual model in trustworthy clouds' infrastructure. Workshop on Cryptography and Security in Clouds, 1–10.
- Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. *Statewide Agricultural Land Use Baseline 2015* (Vol. 1). <http://doi.org/10.1017/CBO9781107415324.004>
- Beck, K. et al. (2001). Manifesto for Agile Software Development. Agile Alliance.
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30–39. <http://doi.org/10.1109/MS.2005.129>
- Caum, C. (2013). Continuous Delivery Vs. Continuous Deployment: What's the Diff? Retrieved from <https://puppetlabs.com/blog/continuous-delivery-vs-continuous-deployment-whats-diff>
- Colavita, F. (2016). Devops movement of enterprise agile breakdown silos, create collaboration, increase quality, and application speed. In *Communications in Computer and Information Science* (Vol. 422, pp. 203–213). http://doi.org/10.1007/978-3-319-27896-4_17
- Davis, J., & Daniels, K. (2015). Effective DevOps. O'Reilly Media, Inc.
- De Bayser, M., Azevedo, L. G., & Cerqueira, R. (2015). ResearchOps: The case for DevOps in scientific applications. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, 1398–1404. <http://doi.org/10.1109/INM.2015.7140503>
- Debois, P. (2011). Devops: A Software Revolution in the Making? *Cutter IT Journal*, 24(8), 1–41.
- Donaldson, L. (2006). The contingency theory of organizational design: challenges and opportunities. In *Organization Design* (Vol. 6, pp. 19–40). http://doi.org/10.1007/0-387-34173-0_2
- Edwards, D. (2014). Introducing DevOps to the traditional enterprise. *InfoQ.com*, (14), 34.
- Erich, F., Amrit, C., & Daneva, M. (2014). Report: DevOps Literature Review. <http://doi.org/10.1007/978-3-319-13835-0>
- Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Continuous delivery. <http://doi.org/10.1007/s13398-014-0173-7.2>
- Hüttermann, M. (2012). Building Blocks of DevOps. In *DevOps for Developers* (pp. 33–47).
- Iden, J., Tessem, B., & Päiväranta, 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. <http://doi.org/10.1016/j.infsof.2010.12.002>
- Jurisch, M. C., Cuno, J., Palka, W., Wolf, P., & Krcmar, H. (2012). An Integrative Model of IT-Enabled Business Process Change: Causal Structures in Theory, Research and Practice. <http://doi.org/10.1109/HICSS.2012.105>
- Jurisch, M. C., Ikaas, C., Palka, W., Wolf, P., & Krcmar, H. (2012). A Review of Success Factors and Challenges of Public Sector BPR Implementations. <http://doi.org/10.1109/HICSS.2012.80>
- Kettinger, W. J., & Grover, V. (1995). Special section: toward a theory of business process change management. *J. Manage. Inf. Syst.*, 12(1), 9–30. <http://doi.org/10.2307/40398151>
- Kim, G. (2011). Top 11 Things You Need to Know About DevOps. IT Revolution Press.
- Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of DevOps. *Lecture Notes in Business Information Processing. Agile Processes, in Software Engineering, and Extreme Programming: 16th International Conference, XP 2015 Helsinki, Finland, May 25-29, 2015 Proceedings*, 212, 212–217. <http://doi.org/10.1007/978-3-319-18612-2>
- Mayer, R., & Benjamin, P. (1995). A framework and a suite of methods for business process reengineering. Idea Group Publishing. <http://doi.org/10.4018/978-1-87828-929-2>
- Moran, J. W., & Brightman, B. K. (2000). Leading organizational change. *Journal of*

- Workplace Learning, 12(2), 66–74.
<http://doi.org/10.1108/13665620010316226>
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *Communications of the ACM*, 48(2), 72–78.
<http://doi.org/10.1145/1060710.1060712>
- Pais, M. (2012). DevOpsDays Italy. Retrieved from
<https://www.infoq.com/news/2012/10/introduce-devops>
- Prodan, M., Prodan, A., & Purcarea, A. A. (2015). Three New Dimensions to People, Process, Technology Improvement Model, 481–490. <http://doi.org/10.1007/978-3-319-16486-1>
- Salehi, F., & Yaghtin, A. (2015). Action Research Innovation Cycle: Lean Thinking as a Transformational System. *Procedia - Social and Behavioral Sciences*, 181, 293–302.
<http://doi.org/10.1016/j.sbspro.2015.04.891>
- Schaefer, A., Reichenbach, M., & Fey, D. (2013). Continuous integration and automation for DevOps. In *Lecture Notes in Electrical Engineering* (Vol. 170 LNEE, pp. 345–358).
http://doi.org/10.1007/978-94-007-4786-9_28
- Swartout, P. (2014). Continuous Delivery and DevOps: A Quickstart guide, Second Edition.
<http://doi.org/10.1017/CBO9781107415324.004>
- Ten Hagen, J., & Heunks, J. (2016). Inbedden van DevOps. *Informatie, Maandblad Voor Informatievoorziening*, 26–33.
- Van Bon, J. (2004). IT Service Management an Introduction based on ITIL. ITSM Library.
- Walls, M. (2013). Building a DevOps Culture. O'Reilly Media, Inc.

