From operational data to business insights

Citation for published version (APA):

Document status and data:
Published: 11/11/2021

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
From operational data to business insights: Adopting data-driven practices in B2B software-intensive companies

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op donderdag 11 november 2021 om 13:30 uur

door

Iris Figalist

geboren te München, Duitsland
Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter: prof.dr. J.J. Lukkien
1e promotor: prof.dr.ir. J. Bosch
Co-promotor: prof.dr. H. Holmström Olsson (Malmö University)
leden: prof.dr. M.G.J. van den Brand
       prof.dr.ir. P.W.P.J. Grefen
       prof.dr. B. Plank (IT University of Copenhagen)
       prof.dr. M.R.V. Chaudron
       dr. C. Elsner (Siemens, Corporate Technology)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.
Abstract

Modern software engineering practices such as DevOps enable product teams to shorten their release cycles by developing, deploying, and operating their products in a continuous loop. Along the way, vast amounts of data are generated throughout the entire product lifecycle from development to deployment to operations. This type of data can play a key role in continuously monitoring and improving products and processes in a data-driven way. The goal of data-driven software engineering is to no longer take decisions based on opinions and gut feelings, instead to rely on and retrieve insights from product and customer data in order to take informed and evidence-based decisions. However, the adoption of data-driven practices within organizations is difficult as it requires a major change of mindset and a large amount of trust and belief in the value of working in a data-driven way.

Moreover, the quality and usefulness of an analysis and its results highly depend on the availability and quality of data. This however, causes uncertainty in applying data-driven approaches in business-to-business (B2B) contexts. The volume of data being generated or even available in B2B contexts is often significantly lower as compared to business-to-customer (B2C) domains. For one, this is caused by a more limited number of customers in B2B contexts. For another, the customer of a software-intensive product is often not the end-user of the product but rather an intermediate entity. As a result, data generated by the end-user is often more complex to analyze or not accessible at all. These limitations often lead to a lack of confidence in the data which in turn makes it difficult to drive the adoption of data-driven practices in B2B software-intensive companies.

This thesis investigates the adoption of data-driven practices in multiple, real-world B2B product teams. The studies are based on case study research conducted across three case companies and twelve product teams. Both qualitative data resulting from interviews and long-term collaboration and quantitative data in the form of a survey were collected. The thesis is structured in three parts addressing three overarching research questions. First, the current state of how product teams in distributed organizations communicate with each other, what kind of information, feedback or data they share, and how they are making and communicating decisions is presented. The second part comprises multiple studies that introduce conceptual models for generating technical and business insights for a variety of roles (e.g. software architects, product managers, or sales) based on operational data. Finally, the third part presents technical, organizational, and people-related challenges in productizing (automating and deploying) prototypical analyses generating insights in that context. A subset of these challenges is further investigated in two additional studies in order to identify and develop approaches for addressing the challenges. The key contributions of this thesis are an overview of conflicting interests and challenges of agile teams in B2B software-intensive companies, four models for systematically generating high-level insights based on low level operational data, a vicious circle highlighting interdependent key drivers impeding the productization of analyses, a framework for the productive use of machine learning in software analytics and business intelligence, and a model for the adoption of a DataOps mindset in B2B software-intensive companies.
Acknowledgments

If someone had told me three and a half years ago what I was getting myself into, I probably would not have believed it. Looking back, I can say that it was an intense and unforgettable journey that I would not have mastered without the help and support of many others.

First of all, I would like to thank my promoter Jan Bosch. Jan has the great gift of pushing people to give their absolute best and achieve things they never thought possible. I am especially grateful for his honesty, unconditional support, and constant feedback, without which this work would never have been achievable.

I also want to thank my co-promoter Helena Holmström Olsson for her continuous support and encouragement. Especially her feedback, which is always right to the point, taught me a lot and helped me to improve the structure and content of my work significantly.

A very special thank you goes to my industrial advisor Christoph Elsner. Christoph is a great mentor with whom I had an uncountable number of valuable and in-depth discussions. It was due to his views and insights that I came up with ideas I wouldn’t have had otherwise. Thank you for the continuous support and feedback.

A warm thanks also goes to my colleagues at Siemens and Siemens Healthineers, especially to Kilian, Johannes, Jan, Martin, Philipp K., Fabian, Thomas, Philipp G., Olli, Marco, and many others that would be too many to list. You have made everyday work a lot of fun, even in the most frustrating times.

Last but not least, I want to thank my family. My parents, Helmut and Claudia, for always supporting and encouraging me; my sisters, Carola and Christina, for teaching me to be resilient and for being the greatest role models; and my husband-to-be, Marius, who has been my rock throughout this entire journey and who always believed in me, even when I didn’t. Thank you, I could not have done it without you.
Contents

List of Figures 9

List of Tables 12

1 Introduction 15
   1.2 Problem Statement and Research Questions 19
   1.3 Research Methodology & Process 20
   1.4 Thesis Overview 27
   1.5 Contribution to the Articles 32
   1.6 Related Publications 34

2 Scaling Agile beyond Organizational Boundaries: Coordination Challenges in Software Ecosystems 35
   2.1 Introduction 36
   2.2 Characteristics of Software Ecosystems 37
   2.3 Research Design 38
   2.4 Results 39
   2.5 Related Work 48
   2.6 Conclusion 50

3 Business as Unusual: A Model for Continuous Real-time Business Insights Based on Low Level Metrics 51
   3.1 Introduction 52
   3.2 Background and Motivation 53
   3.3 Research Method 53
   3.4 Implementation 54
   3.5 Results 62
   3.6 Related Work 65
   3.7 Conclusion 67

4 Customer Churn Prediction in B2B Contexts 69
   4.1 Introduction 69
   4.2 Related Work 70
8.5 Framework Derivation .............................................. 176
8.6 Framework Validation ............................................. 179
8.7 Conclusion ............................................................... 182

9 A Multi-Case Study on Driving the Adoption of a DataOps Mindset in B2B Software-Intensive Companies ........................................... 183
9.1 Introduction ............................................................... 184
9.2 Background ............................................................. 185
9.3 Research Method & Study Design ................................. 187
9.4 Case Study ............................................................... 190
9.5 Findings ................................................................. 208
9.6 Related Work ........................................................... 218
9.7 Threats to Validity .................................................... 220
9.8 Conclusion ............................................................... 220

10 Conclusion ................................................................. 223
10.1 Research Questions .................................................. 223
10.2 Threats to Validity .................................................... 228
10.3 Generalization ......................................................... 230
10.4 Key Contributions .................................................... 231
10.5 Further Research ...................................................... 234

Bibliography .................................................................. 236

Appendix A: Summary .................................................... 253

Appendix B: Curriculum Vitae ........................................... 256
# List of Figures

1-1 Research studies mapped to research questions ........................................ 21
1-2 Incremental research process to establish a deeper understanding of the research field ................................................................. 24
1-3 Step-wise approach to generate high-level insights from low level metrics 28
1-4 Model for efficiently implementing stakeholder-targeted monitoring- and decision-making frameworks .............................................. 29
1-5 Boxplots of web usage metrics across *customer satisfaction classes* .... 30
1-6 Framework for productively applying machine learning ......................... 31

2-1 Actors in software ecosystems ................................................................. 38
2-2 Agile lifecycle ....................................................................................... 40
2-3 Conflicting interests between different ecosystem actors ....................... 42
2-4 Mapping of challenges to different settings ........................................... 45

3-1 Procedure of Case A .............................................................................. 55
3-2 Average duration on pages ................................................................... 56
3-3 Network graph of navigation paths ....................................................... 57
3-4 Navigation paths modeled as Markov chain ......................................... 57
3-5 Website structure optimization ............................................................... 58
3-6 Procedure of Case B .............................................................................. 59
3-7 Platform API usage rates ...................................................................... 60
3-8 Procedure of Case C .............................................................................. 61
3-9 Common operation sequences ............................................................... 61
3-10 Dimensions and process steps of generic model .................................. 63
3-11 Generic model ..................................................................................... 65

4-1 Overview of the mapping approach based on customer phases ............. 75
4-2 Prediction process .................................................................................. 76

5-1 Research Process ................................................................................... 85
5-2 Implementation of Case A ..................................................................... 89
5-3 Implementation of Case B ..................................................................... 92
5-4 Implementation of Case C ..................................................................... 94
5-5 Implementation of Case D ..................................................................... 96
5-6 Implementation of Case E ..................................................................... 98
5-7 Implementation of Case F ..................................................................... 100
5-8 Primary data sources and methods for each stakeholder
5-9 Data and knowledge flow across stakeholders and levels
5-10 Generic process for instantiating monitoring- and decision-making frameworks
5-11 Sharing and reuse of knowledge across stakeholders
5-12 Model for efficient measurement and generation of stakeholder-specific knowledge

6-1 Research process
6-2 Antecedents and consequences of customer satisfaction ([Oliver, 1980], [Parasuraman et al., 1988], [Anderson and Sullivan, 1993], [Eggert and Ulaga, 2002], [Eklof et al., 2018], [Kotler, 2019], [Bolton and Lemon, 1999], [Cronin Jr et al., 2000], [Cronin Jr and Taylor, 1992], [McDougall and Levesque, 2000], [Zeithaml et al., 1996])
6-3 Web usage mining process (from [Srivastava et al., 2000], [Liu, 2011], and [Neelima and Rodda, 2015])
6-4 Model for automated and quantitative customer satisfaction evaluation
6-5 Calculation process of customer satisfaction score
6-6 Mean and median scores of SQMs
6-7 Mean and median dimensional satisfaction scores
6-8 Histogram of customer satisfaction scores
6-9 Boxplots of web usage metrics across customer satisfaction classes

7-1 Research process
7-2 Types of collected data sources
7-3 Rating of adverse factors on the application of data analytics in the respondents’ projects
7-4 Vicious circle consisting of key drivers that prevent a successful realization of AI4SABI
7-5 Identified solutions for the key drivers in the vicious circle

8-1 Research Process
8-2 Framework for productively applying machine learning
8-3 Model deployment in Amazon SageMaker

9-1 Research process
9-2 Analysis results of Case A
9-3 Analysis results of Case D
9-4 Stakeholder phases during the process of adopting a DataOps mindset
9-5 Base model for the adoption of DataOps practices in B2B software-intensive companies
9-6 Instantiation of model in stakeholder phase I (skepticism)
9-7 Instantiation of model in stakeholder phase II (interest)
9-8 Instantiation of model in stakeholder phase III (exploration / experimentation)
9-9 Instantiation of model in stakeholder phase IV (collaboration)
List of Tables

1.1 Overview of objective, research method, and result type of each research study .......................... 24
1.2 Overview of publications ........................................ 25
1.3 Overview of case companies, products, and participants ................................. 26

2.1 Description of ecosystems .................................................. 38
2.2 Description of conflicting interests between keystone and partners and the influencing factors in different phases .............................. 43
2.3 Description of coordination challenges caused by divergent interests of actors within the ecosystem ................................. 46

3.1 Most common navigation paths .............................................. 56
3.2 Common platform API combinations per cluster .......................... 59

4.1 Extracted features per type ................................................... 73

5.1 Overview of cases and project contexts ...................................... 83

6.1 Definition summary of the SaaS-QUAL factors [Benlian et al., 2011] . 116
6.2 Evolution and comparison of relevant service quality measurement scales ([Parasuraman et al., 1988], [Kettinger and Lee, 2005], [Parasuraman et al., 2005], [Swaid and Wigand, 2009], [Benlian et al., 2011]) .... 117
6.3 Customer satisfaction measurement methods ................................ 117
6.4 Overview of web usage metrics [Zumstein, 2012] ............................ 119
6.5 Calculated web usage metrics of 1400 customers ............................ 129

7.1 Interviewees grouped by products and function ............................... 140
7.2 Overview of expert interviewees ............................................. 144
7.3 Overview of sample size, respondents and experience levels ................. 149
7.4 Tasks performed by the participants ......................................... 149
7.5 Using results for decision making, productizing analysis and setting up an infrastructure ................................. 151
7.6 Rating of adverse factors on the application of data analytics in the respondents’ projects; the bold numbers indicate which factors were ranked as either very (at least 30%) or extremely (at least 10%) influential by the respondents ................................. 151
7.7 Key drivers and challenges derived from interview study (I) and survey (S) ........................................ 153
7.8 Identified solutions for key driver: Low data quality ................. 159
7.9 Identified solutions for key driver: Inability to cross cultural gap . 160
7.10 Identified solutions for key driver: Ineffective prototypical analysis . 160
7.11 Identified solutions for key driver: Inability to prove value .......... 161
7.12 Identified solutions for key driver: Lack of priority, time and resources 161

8.1 Common activities in data management and processing for machine learning .................................................. 171
8.2 Common challenges in data management and processing ............ 172
8.3 Common activities in model preparation, building, and evaluation .... 173
8.4 Common challenges in model preparation, building, and evaluation . 174
8.5 Common activities in model deployment and model serving .......... 175
8.6 Common challenges in model deployment and model serving ......... 176

9.1 Overview of case products and stakeholders participating in the study 188
9.2 Summary of Case A (○ = data scientist/data engineer activity, △ = stakeholder activity) ........................................ 195
9.3 Summary of Case B (○ = data scientist/data engineer activity, △ = stakeholder activity) ........................................ 199
9.4 Summary of Case C (○ = data scientist/data engineer activity, △ = stakeholder activity) ........................................ 203
9.5 Summary of Case D (○ = data scientist/data engineer activity, △ = stakeholder activity) ........................................ 207
9.6 Summary of activities across all cases (○ = data scientist/data engineer activity, △ = stakeholder activity) ...................... 208
Chapter 1

Introduction


1.1.1 Agile and DevOps

Nowadays, the use of agile methods has become an integral part of modern software development. Agile practices are being applied by the majority of development teams in software-intensive organizations [Hemon et al., 2020]. The focus of software-intensive companies lies on the production of software, for instance of platforms, applications, tools, or embedded systems. Emerging from the need of developing software in a more flexible way, agile methodologies allow software development teams to shorten their deployment and release cycles [Cockburn, 2006], [Dingsøyr et al., 2018], [Dybå and Dingsøyr, 2008], [Hemon et al., 2020] and to react to changing customer needs in a quick and timely manner [Cohen et al., 2004], [Beck et al., 2001].

With the focus of agile methods being mostly on software development activities, operations who are handling the release of the software soon became a bottleneck within the product lifecycle [Hemon et al., 2020]. In order to enable a continuous loop between development and operations, the concept of DevOps emerged and soon gained prominence in a large number of companies moving from agile towards DevOps [Hemon et al., 2020].

"DevOps is a development methodology aimed at bridging the gap between Development (Dev) and Operations (Ops), emphasizing communication and collaboration, continuous integration, quality assurance, and delivery with automated deployment utilizing a set of development practices" [Jabbari et al., 2016]. The underlying concept of DevOps is to bring individual teams, especially development and operations, closer together in order to develop and deploy resilient and high-quality software in short release cycles [Dyck et al., 2015], [Hüttermann, 2012], [Penners and Dyck, 2015]. Key activities and principles of DevOps are culture, automation, measurement, and sharing [Humble and Molesky, 2011], [Hüttermann, 2012]. As a result, transitioning towards DevOps involves both technical and cultural changes to the ways teams and organizations are being used to work [Erich et al., 2017], [Leite et al., 2019].
Technical changes include, among others, treating infrastructure as code, automating deployments using build and CI/CD tools, and continuously monitoring infrastructure and system behavior in production [Ebert et al., 2016] [Erich et al., 2017] [Lwakatare et al., 2016]. On the organizational side, it is critical to foster straightforward communication and shared responsibilities within and across teams in order to build and strengthen a collaborative culture based on trust [Luz et al., 2018] [Walls, 2013].

The advantages entailed by a successful transformation towards DevOps are manifold. On the one side, features are being implemented faster and released more frequently and the high level of automation leads to an improved quality assurance. On the other side, the increased interaction across teams enhances their collaboration and communication which ultimately creates more efficiency in the daily work routine [Hüttermann, 2012], [Riungu-Kalliosaari et al., 2016]. However, there are several challenges impeding a successful transformation towards DevOps. It is extremely difficult to change the ways people are being used to work [Hüttermann, 2012]. Therefore, key challenges for the adoption of DevOps are an insufficient communication and a "deep-seated company culture" [Riungu-Kalliosaari et al., 2016] which does not align with the DevOps mindset and requires organizations, teams, and individuals to rethink their current roles and processes. Moreover, its applicability highly depends on the "industry constraints and feasibility in different domains" [Riungu-Kalliosaari et al., 2016]. In B2B contexts, software products are not purchased by individuals but by other companies operating in various, sometimes even safety-critical domains. Therefore, continuous delivery of these software products is often not feasible because avoiding downtime can be a critical factor for customers [Rissanen and Münch, 2015b]. The production environment is often owned by the customer instead of the product provider impeding an automated deployment of software products in the B2B domain. Oftentimes, customers need to be trained to use a certain product in the B2B domain which leads to a negative attitude towards new features and frequent updates [Rissanen and Münch, 2015b]. On top of that, it is very difficult to collect customer feedback of B2B software-intensive products since there is usually an intermediate entity (e.g. company representative or purchasing manager) between product provider and end-user hindering the use of direct feedback channels [Rissanen and Münch, 2015a].

1.1.2 Data-driven Software Engineering and Data-driven Decision Making

Although software providers often believe they know what their consumers want, they frequently make erroneous assumptions about what truly creates value to their customers. In addition to that, gaining a comprehensive understanding of all customers can be difficult and time-consuming because each consumer values different parts or features of the product [Butz Jr. and Goodstein, 1996]. Many of the existing approaches are based on a limited number of subjective opinions (e.g. from customer interviews) [Marciuska et al., 2014]. However, with the adoption of DevOps
and continuous deployment practices, customer perceptions can change rapidly with every new release. This makes it difficult to track the customers' perceptions of the product through qualitative studies in a continuous way.

With an increasing amount of data being available and with practices such as DevOps being implemented, there is the opportunity to make use of data, learn from it and include it as a basis for decisions making. As a result, many companies started to adopt data-driven software engineering practices, especially in B2C and software-as-a-service (SaaS) contexts. The concept behind data-driven software engineering is to collect quantitative data (e.g. product usage data or demographic customer data) to learn more about how a product is being used by customers and to take evidence-based decisions based on data instead of opinions and gut feelings [Olsson and Bosch, 2014], [Olsson and Bosch, 2019]. Related fields that foster the principles of data-driven software engineering are software analytics [Buse and Zimmermann, 2012] [Menzies and Zimmermann, 2013], business intelligence [Negash and Gray, 2008], or online controlled experiments [Kohavi et al., 2013] [Kohavi and Longbotham, 2017].

The main advantage of data-driven software engineering is that it closes the gap between customer feedback and product development [Olsson and Bosch, 2014]. It can be a powerful asset to drive the evolution of software-intensive products and to invest resources in a more targeted manner [Olsson and Bosch, 2019].

Data-driven decision making is a key component in data-driven software engineering. It "refers to the practice of basing decisions on the analysis of data rather than purely on intuition" [Provost and Fawcett, 2013]. By learning more about how a software product is being used by customers in practice, strategic decisions on product evolution, feature prioritization, and concepts for increasing customer value can be made and evaluated based on data [Olsson and Bosch, 2014]. Previous studies indicate that companies that continuously assess customer value by analyzing and making decisions based on customer and product data often become more profitable over time and experience an increased productivity and market value [Fabijan et al., 2017b], [Provost and Fawcett, 2013].

In order to promote data-driven decision making in software-intensive companies, data-analytical thinking is "important not just for the data scientist but throughout the organization" [Provost and Fawcett, 2013]. A successful adoption of data-driven practices requires fundamental changes to the way people are being used to work since decisions are now made based on data instead of subjective views and opinions. To achieve this, companies like Microsoft started to bring in data scientists into traditional software development teams in order to foster a close cooperation between the two [Kim et al., 2016] [Kim et al., 2017]. As their ways of working often differ significantly, there is a risk for emerging conflicts between data scientists and other members of the software development team. It is particularly challenging to convey the value of an analysis and its results to colleagues, managers, and other stakeholders. Therefore, all team members need to be willing to engage in a close and continuous collaboration to, for instance, jointly define use cases, go over the data, and discuss and interpret analysis results [Kim et al., 2016].

One of the key challenges during the adoption of data-driven practices is to identify use cases and analyses that turn into actionable insights. If results are not actionable
or cannot be used in practice, the value of an analysis will not be visible [Yang et al., 2017]. The information needs of different roles involved in product development are not equally well-researched. While the needs of software developers, for example, are already being studied in great detail, there is a lack of research on the information needs of other stakeholders such as project or product management, sales, or marketing [Abdellatif et al., 2015], [Hassan et al., 2013]. This ultimately causes a gap between operational data that is being collected and the generation of business insights for stakeholders that are not well understood.

Moreover, there are additional challenges directly related to the collected data. The quality and availability of data is a determining factor for the quality and usefulness of an analysis and its results [Gudivada et al., 2017] [Polyzotis et al., 2018]. Precisely this often causes uncertainty in applying data-driven approaches in B2B contexts. Oftentimes, the volume of data being generated or even available in B2B contexts is significantly lower compared to B2C domains [Bohanec et al., 2017]. For one, this is caused by a more limited number of customers in B2B contexts which is a typical characteristic of B2B businesses [Russo et al., 2016]. For another, the customer of a software-intensive product is often not the end-user of the product but rather an intermediate entity [Rissanen and Münch, 2015a]. As a result, data generated by the end-user is often more complex to analyze or not accessible at all. These limitations often lead to a lack of confidence in the data which in turn makes it difficult to drive the adoption of data-driven practices in B2B software-intensive companies [Walls, 2013].

1.1.3 DataOps

Nowadays, it is possible to collect, store, and analyze data at every step of the DevOps lifecycle. This enables practitioners to retrieve data-driven feedback on a continuous basis [Riungu-Kalliosaari et al., 2016]. With increasing amounts of data to be stored, processed, and analyzed, the concept of DataOps emerged as an extension of DevOps [Capizzi et al., 2019]. DataOps can be defined as "a set of practices, processes, and technologies that combines an integrated and process-oriented perspective on data with automation and methods from agile software engineering to improve quality, speed, and collaboration and promote a culture of continuous improvement" [Ereth, 2018]. While it can be considered as a facilitator for data analytics and machine learning, the shift towards DataOps entails adjustments and new ways of working not only on a technological level, but also on a cultural and organizational one [Ereth, 2018]. Organizational changes are considered to be a difficult undertaking because they necessitate a shift in mindset among all individuals within an organization and across hierarchies. Because data does not always fit everyone’s beliefs and expectations, the transition to data-driven ways of working is frequently met with rejection and resistive behavior, ultimately resulting in a lack of acceptance [Olsson and Bosch, 2019]. Data should also not be trusted blindly without questioning, in fact it is very important to understand the explanatory power and its limitations. Changing mindsets and building a new culture inside a company is a constant process that is dependent on trust, excitement, and belief in the new way of working [Walls, 2013]. As a result, it
is critical to enable stakeholders from other fields than data science to engage with data and properly comprehend and interpret it in the right way [Ereth, 2018]. It is important to gradually establish this trust and to effectively demonstrate the necessity for an organizational transformation and its advantages [Walls, 2013].

On a technological side, data science projects often start out as prototypical implementations to try out new ideas and get first results fast [Provost and Fawcett, 2013], [Sculley et al., 2015]. A major downside of this is that the results of small-scale prototypes often do not reflect the results in reality [Sculley et al., 2015]. On the one hand, prototypes are a strong tool for explaining and conveying ideas to stakeholders, since an analysis becomes more concrete when some preliminary results are available. On the other hand, a large part of analyses generate the expected value only when performed in an automated and continuous manner. However, the conversion of prototypical analyses into automated and deployed analyses is complex and time-consuming [Misirli et al., 2013], [Sculley et al., 2015]. This results in "prototypes or early models that lack the final deployment step, and hence, any real-world impacts" [Misirli et al., 2013].

1.2 Problem Statement and Research Questions

1.2.1 Problem Statement

Based on the background presented in the previous section, the following problem statement is derived as a starting point of the thesis:

_Due to ever increasing possibilities to collect, store, and analyze data at every step of the product development lifecycle, data-driven software engineering practices are becoming an integral part in many software-intensive companies, especially in the B2C and online domain. However, the transition towards a data-driven way of working remains complex as fundamental changes to existing decision-making processes and the way people are being used to work are unavoidable. It is very challenging to establish the necessary mindset throughout an organization since it requires a certain level of trust and confidence in the data and in working in a data-driven way. In B2B contexts, this often results particularly difficult as companies are often more traditional and more reluctant to change as compared to companies in B2C domains, for instance due to their specialized customer requirements (e.g. for safety-critical systems). Moreover, the information needs of different stakeholders need to be well-understood in order to generate the expected value. Nonetheless, existing literature on the information needs of non-technical stakeholders (e.g. product management or sales) is quite scarce. The analyses implemented for addressing stakeholder-targeted information needs often remain in a prototypical stage and are not being operationalized (e.g. automated and deployed). As a result, these analyses can often not provide the desired value to address the stakeholders’ information needs. In addition to that, there are B2B-specific data characteristics further impeding the application of data-driven practices in B2B software-intensive companies. Specifically, lower volumes of data and the differentiation between customers and end users complicate the collection_
and analysis of data. Consequently, a better understanding of all stakeholders involved in the product development lifecycle is required as well as methods and techniques for 1) generating stakeholder insights based on operational data while taking B2B-specific data characteristics into consideration, for 2) operationalizing these analyses and for 3) driving the adoption of a data-driven mindset throughout teams in B2B software-intensive companies.

1.2.2 Research Questions

The following three research questions were derived from the problem statement in Section 1.2.1. They constitute the guiding questions behind the individual research studies presented in this thesis.

- RQ1: What are the key challenges that teams in B2B software-intensive companies face when coordinating their efforts, making decisions, and exchanging data and feedback?

- RQ2: How can operational data with B2B-specific data characteristics be leveraged to generate technical and business insights for different roles involved in the product lifecycle?

- RQ3: What are the challenges on a technical, organizational, and individual level in operationalizing analyses for actual use in practice and how can these challenges be addressed?

This thesis consists of multiple research studies which either fully or partly address one of the research questions. Figure 1-1 visualizes how the studies are connected to each other (indicated by arrows) and how each study can be mapped to the defined research questions (indicated by color and label RQ1-RQ3). The striped arrows indicate a purely chronological relationship while the continuous arrows represent a direct dependency between studies in the sense that the output of one study was used as a motivation or direct input for a follow-up study.

1.3 Research Methodology & Process

1.3.1 Research Methodology

Selecting appropriate methods for empirical software engineering research is a complex task as "software engineering is a multi-disciplinary field, crossing many social and technological boundaries" [Easterbrook et al., 2008]. Each research methodology typically serves a specific purpose. Based on Robson’s [Robson, 2002] classification, Runeson and Höst [Runeson and Höst, 2009] differentiate between four types of purposes:

- **Exploratory:** finding out what is happening, seeking new insights, and generating ideas and hypotheses for new research
Figure 1-1: Research studies mapped to research questions

- **Descriptive**: portraying a situation or phenomenon
- **Explanatory**: seeking an explanation of a situation or a problem, mostly but not necessary in the form of a causal relationship
- **Improving / prescriptive**: trying to improve a certain aspect of the studied phenomenon

The objectives of the research studies presented in this thesis cover all four types of purposes. In the beginning of the research project, the studies are mostly of an exploratory and descriptive nature while this shifts to descriptive, explanatory, and prescriptive towards the end of the thesis.

**Research Methods in Software Engineering**

Easterbrook et al. [Easterbrook et al., 2008] identified five classes of research methods that they consider relevant to software engineering: **Controlled experiments, case studies, survey research, ethnographies, and action research.** These classes can be defined as follows:

- **Controlled experiments**: "A controlled experiment is an investigation of a testable hypothesis where one or more independent variables are manipulated to measure their effect on one or more dependent variables. Controlled experiments allow us to determine in precise terms how the variables are related and, specifically, whether a cause-effect relationship exists between them." [Easterbrook et al., 2008]
• Case study: "An empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident." [Yin, 2014]

• Survey research: "Survey research is used to identify the characteristics of a broad population of individuals. [...] The defining characteristic of survey research is the selection of a representative sample from a well-defined population, and the data analysis techniques used to generalize from that sample to the population, usually to answer base-rate questions." [Easterbrook et al., 2008]

• Ethnographies: "Ethnography is a form of research focusing on the sociology of meaning through field observation. The goal is to study a community of people to understand how the members of that community make sense of their social interactions." [Easterbrook et al., 2008]

• Action research: "In Action Research, the researchers attempt to solve a real-world problem while simultaneously studying the experience of solving the problem." [Easterbrook et al., 2008]

In addition to that, there are mixed-methods approaches that are based on the collection and analysis of both qualitative and quantitative data, either within one or multiple research studies [Easterbrook et al., 2008]. The research studies presented in this thesis are all based on case study research, survey research and a mixed-methods approach, resulting in empirical models derived from the observations. For case study research, there exist three different types of data collection [Lethbridge et al., 2005]:

• Direct methods: Researcher is in direct contact with the subjects to collect data in real time (e.g. interviews)

• Indirect methods: data is collected without directly interacting with subjects (e.g. video recordings)

• Independent analysis: already available work artifacts are analyzed (e.g. requirements specification, failure reports)

Moreover, empirical research in software engineering can either be conducted in an inductive and deductive way. In inductive research studies, "the researcher first observes with an open mind, identifies patterns in the observation, sets up tentative hypotheses, and finally relates them to existing theory or develops new theory" [Runeson et al., 2012]. Deductive research approaches start by building hypotheses from existing theories which are then either confirmed or rejected based on observations [Runeson et al., 2012].

**Types of Results in Software Engineering Research**

According to Shaw [Shaw, 2002] there exist eight different types of results in software engineering research:

---

22
1.3.2 Research Process

One of the first steps in the research process is to define the research questions that are to be addressed in the study. Based on this, an appropriate research method can be selected. In early stages of an overall research process, these questions tend to be very exploratory [Easterbrook et al., 2008]. This also applies to the research studies presented in this thesis. The earlier studies (Chapters 2-6) are of an explorative and descriptive nature, and there is a shift towards descriptive and prescriptive studies towards the end of the thesis. While a research problem can typically be addressed by one or more different research methods, the selection of methods depends on multiple factors, such as the access to resources or how closely a method is aligned with the research questions [Easterbrook et al., 2008].

Three overarching research questions were defined in Section 1.2.2 in order to address the problem statement presented in Section 1.2.1. Each research question is being addressed by one or more research studies in this thesis. Figure 1-2 highlights the research process and the order of research studies conducted to incrementally establish a deeper understanding of the research field.

Each study contains its own research design. The different objectives, research methods, and results types of the research studies are highlighted in Table 1.1. Table 1.2 gives an overview of the publications that each of the chapters is based on.

The majority of research studies presented in this thesis are based on case study research. In addition to that, there is one study combining case study research and survey research in a mixed-methods approach. For conducting the case studies, the guidelines by Runeson and Höst were followed who propose the following process steps: 1. Case study design: objectives are defined and the case study is planned;
2. Preparation for data collection: procedures and protocols for data collection are defined; 3. Collecting evidence: execution with data collection on the studied case; 4. Analysis of collected data; and 5. Reporting.

The case studies presented in this thesis were conducted in three different case companies, Company A, Company A1 (subsidiary of Company A), and Company B. Company A is a large, industrial company with around 293,000 employees and multiple subsidiary companies. The company and its subsidiaries offer a diverse product portfolio in a variety of different domains, such as industry, infrastructure, mobility, healthcare (Company A1), and energy. It is an international company with product teams that are spread all over the world (mostly in Europe, USA, India, and China). The investigated products and product teams in Company A were 1) an industrial platform provider (I) hosting company-internal and external applications for industrial device management and analysis; 2) an industrial platform provider (II) hosting applications for device configurations, implementation of device logic and commissioning of industrial devices including their networking; 3) a remote connection platform providing services to establish remote connections to industrial devices and plants;
Table 1.2: Overview of publications

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Published as / submitted to</th>
</tr>
</thead>
</table>

and 4) two industrial application hosted on industrial platform I. Company A1 is a platform provider established in the healthcare domain. Several platform-internal and external medical applications are hosted on the platform. Over the course of this thesis the platform and three of its applications were investigated in multiple research studies. Finally, Company B is a platform provider operating in the advertising domain. The platform offers a variety of functionalities to create and monitor advertising campaigns and to create and export reports and dashboards of the campaigns.

In total, 29 individuals distributed across twelve independent product teams participated in one or more of the case studies. The participants cover ten different types of roles which were investigated over the course of this thesis: product managers, product owners, software architects, operation engineers (who also take on data science tasks), sales representatives, digital solution experts, demand managers, operation managers, chief technology officers (CTO), and vice presidents (VP) of product. Table 1.3 gives an overview of the case companies, products and product teams, and participants. Six of the eight studies in this thesis were conducted in an inductive manner, starting from observations to identifying patterns to deriving procedures and models. The remaining two studies (Chapter 6 and Chapter 8) were conducted in a deductive way by building hypotheses from existing theories that are then either confirmed or rejected based on observations.

The collected data was obtained by either direct methods (e.g. interviews) or independent analysis (e.g. operational data), sometimes combining both data collection techniques in the same study. As a result, the type of collected data comprises both qualitative (interview results) and quantitative data (survey and measurements).
Quantitative data was analyzed using statistics, and thematic coding [Maguire and Delahun, 2017], [Robson and McCartan, 2016] and categorization and sorting [Runeson and Höst, 2009] was used to analyze qualitative data.

The interviews conducted as part of the research studies were either designed in a semi-structured or unstructured manner. In unstructured interviews, general concerns and interests are formulated by the researcher but the direction in which a conversation can develop is not predetermined. For semi-structured interviews, a set of questions is prepared by the researcher but the conversation decides in which order the questions are handled [Runeson and Höst, 2009].

There are four different types of triangulation which can be used to increase the validity of results [Robson and McCartan, 2016]:

1. **Data triangulation**: The use of more than one method of data collection (e.g. observation, interviews, documents)

2. **Observer triangulation**: Using more than one observer in the study

3. **Methodological triangulation**: Combining qualitative and quantitative approaches

4. **Theory triangulation**: Using multiple theories or perspectives.

By following the guidelines from Runeson and Höst [Runeson and Höst, 2009] and Robson and McCartan [Robson and McCartan, 2016], the research studies presented in this thesis were primarily validated based on a combination of data triangulation,
methodological triangulation and observer triangulation. For example, a comparison of literature, semi-structured interviews, and quantitative data (e.g. survey results, measurements) were used for data triangulation in multiple studies. Moreover, a combination of qualitative (semi-structured interviews) and quantitative approaches (survey) was applied for a methodological triangulation in Chapter 7. Interviews were partially attended by multiple researchers and interview transcripts and results were always discussed among a group of at least four researchers in order to achieve observer triangulation.

1.4 Thesis Overview

This thesis consists of multiple research studies which either fully or partly address one of the research questions raised in Section 1.2.2. The studies presented in Chapters 4, 5, 6, 8, and 9 include the implementation of code for concrete use cases (e.g. customer churn). However, the topics of these use cases are often independent of the overarching goal of the study since the focus lies more on the identification of patterns and methodologies irrespective of the concrete use case. The following subsections describe how each research study contributes to its corresponding research question.

1.4.1 Research Studies Addressing RQ1

The first research question (What are the key challenges that teams in B2B software-intensive companies face when coordinating their efforts, making decisions, and exchanging data and feedback?) was addressed in one research study: Investigate current status of inter-team coordination, exchange of data and feedback, and decision-making processes (presented in Chapter 2).

The study presents an interview-based case study of two industry cases in which the current communication structures, exchange of data and feedback, and decision-making processes were investigated. The results highlight a number of conflicting interests between different actors, which lead to different types of coordination challenges of teams in B2B software-intensive companies. For example, there is a lack of established processes to collect and analyze data which makes it difficult to obtain a broad picture of all customers.

1.4.2 Research Studies Addressing RQ2

Research question RQ2 (How can operational data with B2B-specific data characteristics be leveraged to generate technical and business insights for different roles involved in the product lifecycle?) is addressed by four research studies in this thesis:

- Identify information needs of product managers and describe process for measuring high-level business insights based on low-level operational data (presented in Chapter 3)
- Investigate how to deal with B2B-specific data characteristics based on a customer churn prediction use case (presented in Chapter 4)

- Identify information needs of various stakeholders involved in product development and propose concept for implementing stakeholder-targeted monitoring-and decision-making frameworks (presented in Chapter 5)

- Analyze relationship between customer satisfaction and service quality & web usage metrics in B2B contexts (presented in Chapter 6)

The first study conducted to identify information needs of product managers and describe a process for measuring high-level business insights based on low-level operational data builds on parts of the findings of RQ1 (lack of established processes to collect and analyze data which makes it difficult to obtain a broad picture of all customers). The case study describes the collaboration with product managers in three industry cases. During the collaboration, the individual information needs of the product managers were assessed and a process for generating high-level business insights based on low-level operational data was designed and implemented for each case. In one case, for instance, the API usage rates of the customer applications were analyzed. First, some basic statistics were generated to get an overview of the data. Building upon this, a clustering technique was applied to the data set in order to identify API usage patterns. The clustering results were then used as input to train a prediction model for recommending APIs to the platform’s end users. Figure 1-3 gives an overview of the step-wise procedure. The result of the case study is a generic model for generating high-level business insights while incrementally increasing their complexity by using previous outputs as inputs for further analyses.

![Figure 1-3: Step-wise approach to generate high-level insights from low level metrics](image)

The second study addressing RQ2 investigates how to deal with B2B-specific data characteristics based on a customer churn prediction use case. The study presents an approach for mapping customer-generated data to end-user-generated data. In addition to that, a process is presented for how the mapped data can be used to predict
customer churn in B2B contexts. The third study addressing RQ2 of this thesis identifies information needs of various stakeholders in software-intensive companies and proposes concept for implementing stakeholder-targeted monitoring- and decision-making frameworks. It is based on the previous two studies that already addressed product manager and sales representatives. A case study was conducted in six industry cases covering four different types of stakeholders (product management, sales, software architects, and operations engineers). The information needs of the stakeholders were iteratively assessed during multiple interviews. Based on their interests, a monitoring- and decision-making framework was implemented for each stakeholder. The results of the case study indicate that the implementation of such frameworks can be made more efficient by focusing on a subset of data sources related to the stakeholders’ information need and by sharing and reusing existing components across stakeholders. A model for efficiently implementing monitoring- and decision-making frameworks was derived from the findings of the case study. Key aspects of the model are iterative feedback and specification cycles as well as the reuse of appropriate components to speed up the instantiation process and maximize efficiency, as indicated in Figure 1-4.

![Figure 1-4: Model for efficiently implementing stakeholder-targeted monitoring- and decision-making frameworks](image)

The fourth study analyzes the relationship between customer satisfaction and service quality and web usage metrics in B2B contexts. Closer investigating B2B-specific data characteristics, this study is related to the second study of this research question. Following a deductive research approach, a model for measuring customer satisfaction on B2B online platforms was derived from existing literature. It is based on domain-specific service quality and web usage metrics and enables a continuous measurement of customer satisfaction without requiring active customer participation. Customer
satisfaction scores can be calculated using a customer satisfaction score formula based on different service quality metrics derived from existing literature. Based on their individual score, customers are grouped into customer satisfaction classes. In order to cross-validate the results, a classification model is trained using the customers’ web usage metrics. The applicability of the model was validated by instantiating it in a real-world B2B online platform. The majority of customers were assigned to the classes *neither satisfied nor dissatisfied* (class 2) or *satisfied* (class 3). Figure 1-5 highlights the difference in usage behavior across the two customer satisfaction classes. Ultimately, the classification model that was trained based on the web usage metrics achieved an accuracy of 91.2%.

1.4.3 Research Studies Addressing RQ3

The last research question RQ3 (*What are the challenges on a technical, organizational, and individual level in operationalizing analyses for actual use in practice and how can these challenges be addressed?*) is addressed by the following three research studies:

- **Identify challenges of productizing AI-based analytics for use in practice and investigate solutions for addressing the challenges** (presented in Chapter 7)

- **Describe end-to-end process for implementing and deploying ML-based software analytics and business intelligence solutions** (presented in Chapter 8)

- **Investigate how to drive the adoption of a data-driven mindset in B2B software-intensive companies** (presented in Chapter 9)

After providing multiple analyses to the product teams participating in the studies addressing RQ2, the teams struggled to actually productize (e.g. automate and deploy) the analyses. Therefore, the first study of RQ3 identifies challenges of productizing AI-based analytics for their use in practice and investigates solutions for addressing the identified challenges. In a first step, a case study was designed to identify the challenges in transforming a prototypical AI-based analysis into an automated and deployed analyses. The case study consisted of both a qualitative interview study...
and a quantitative survey. The results of the case study indicate that there are inter-related key drivers that form a vicious circle impeding the productization of analyses. Based on this, a set of potential solutions was extracted from existing literature as well as from the interview and survey results. These solutions were validated by a focus group of experts.

The remaining studies covered within this thesis build on the challenges identified in the previous study. In the context of software analytics and business intelligence, for example, the previous study indicated that many teams experience a lack of expertise in data engineering, data analytics, and in building infrastructure for data analytics. Therefore, the second study describes an end-to-end process for implementing and deploying ML-based software analytics and business intelligence solutions. The process was derived in a deductive manner from existing literature and was structured into three iterative cycles representing different stages in a ML model’s lifecycle: prototyping, deployment, and update (blue, green, and orange cycles in Figure 1-6). The goal of this framework is to specifically support the transitions between the stages while also covering all important activities from data collection to retraining deployed ML models. In order to validate the applicability of the framework in practice, it was compared to and applied to a real-world software analytics and business intelligence solution.

![Figure 1-6: Framework for productively applying machine learning](image)

Since the first study highlighted that the mindset towards data-driven ways of working is often a huge challenge, the last study of this thesis investigates how to drive the adoption of a data-driven mindset in B2B software-intensive companies. This case study describes the long-term collaboration with four independent product teams.
that are in the process of driving the adoption of a data-driven mindset. The results indicate the importance of adapting different steps in the adoption process to the stakeholders’ needs and mindsets. Four stakeholder phases were identified in the case study and each phase requires a different type of interaction between stakeholders and data engineers and data scientists. In addition to that, a model for incrementally establishing a data-driven mindset was derived by combining data-driven practices applicable to B2B-specific data characteristics with respective stakeholder interactions. The models derived from the studies presented in Chapter 8 and Chapter 9 can be used to address some of the challenges identified in Chapter 7.

1.5 Contribution to the Articles

This thesis contains articles resulting from research and work of multiple authors. This section describes the personal contributions to each of the articles using the taxonomy of Elsevier’s CRediT author statement.

Chapter 2: Scaling Agile beyond Organizational Boundaries: Coordination Challenges in Software Ecosystems (Iris Figalist, Christoph Elsner, Jan Bosch, Helena Holmström Olsson)

Iris Figalist: Conceptualization, methodology, investigation, data curation, writing (original draft and editing); Christoph Elsner: Conceptualization, supervision, writing (review); Jan Bosch: Conceptualization, supervision, writing (review); Helena Holmström Olsson: Conceptualization, supervision, writing (review)

Chapter 3: Business as Unusual: A Model for Continuous Real-time Business Insights Based on Low Level Metrics (Iris Figalist, Christoph Elsner, Jan Bosch, Helena Holmström Olsson)

Iris Figalist: Conceptualization, methodology, investigation, software, data curation, writing (original draft and editing); Christoph Elsner: Conceptualization, supervision, writing (review); Jan Bosch: Conceptualization, supervision, writing (review); Helena Holmström Olsson: Conceptualization, supervision, writing (review)

Chapter 4: Customer Churn Prediction in B2B Contexts (Iris Figalist, Christoph Elsner, Jan Bosch, Helena Holmström Olsson)

Iris Figalist: Conceptualization, methodology, investigation, software, data curation, writing (original draft and editing); Christoph Elsner: Conceptualization, supervision, writing (review); Jan Bosch: Conceptualization, supervision, writing (review); Helena Holmström Olsson: Conceptualization, supervision, writing (review)

1https://www.elsevier.com/authors/policies-and-guidelines/credit-author-statement
1.6 Related Publications

One related publication is not included in this thesis:

Figalst, I., Elsner, C., Bosch, J., & Olsson, H. H. (2020, August). Breaking the Vicious Circle: Why AI for software analytics and business intelligence does not take off in practice. In 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 5-12). IEEE. This conference paper is a shorter presentation of parts of the work described in Chapter 7.
Chapter 2

Scaling Agile beyond Organizational Boundaries: Coordination Challenges in Software Ecosystems


Abstract

The shift from sequential to agile software development originates from relatively small and co-located teams but soon gained prominence in larger organizations. How to apply and scale agile practices to fit the needs of larger projects has been studied to quite an extent in previous research. However, scaling agile beyond organizational boundaries, for instance in a software ecosystem context, raises additional challenges that existing studies and approaches do not yet investigate or address in great detail. For that reason, we conducted a case study in two software ecosystems that comprise several agile actors from different organizations and, thereby, scale development across organizational boundaries, in order to elaborate and understand their coordination challenges. Our results indicate that most of the identified challenges are caused by long communication paths and a lack of established processes to facilitate these paths. As a result, the participants in our study, among others, experience insufficient responsivity, insufficient communication of prioritizations and deliverables, and alterations or loss of information. As a consequence, agile practices need to be extended to fit the identified needs.
2.1 Introduction

Agile practices in software development have been around for quite some time and originally emerged due to the need to adapt faster to changing customer requirements [Cohen et al., 2004]. Developing in iterations provides the required flexibility and enables agile projects to "identify and respond to changes more quickly than a project using a traditional approach" [Cohen et al., 2004]. With the focus being on "individuals and interactions over processes and tools" [Alliance, 2001], preferably through face-to-face communication, agile practices initially aimed at small, local teams. However, as agile development became more popular, larger organizations adapted certain practices as well [Jørgensen, 2018]. As a result, large-scale agile frameworks, such as SAFe [Leffingwell, 2016] and LeSS [Larman and Vodde, 2016], were developed to provide guidance to large organizations. The application of scaled agile methods has already been investigated to a great extent, e.g. in [Jørgensen, 2018], [Dingsøyr et al., 2018], [Bjørnson et al., 2018], [Begel et al., 2009], [Bick et al., 2018], [Rautiainen et al., 2011], and [Uludağ et al., 2018]. The focus, however, lies mostly on large-scale projects or multiteams within one organization. This lead us to the question: How to scale agile even further, beyond organizational boundaries?

Software ecosystems can be defined as "the interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services. [...]" [Manikas and Hansen, 2013]. Opening up a platform to external developers enables platform operators to expand their offerings and provide further functionalities that they would not have been able to develop themselves, thereby providing more value to the customer but at the same time requiring additional coordination efforts [Bosch, 2009], [Bosch and Bosch-Sijtsema, 2010].

In large-scale distributed, agile teams, inter-team coordination has previously been identified as a major challenge [Begel et al., 2009]. As large-scale software ecosystems differ from traditional organizations in various ways, these challenges cannot directly be adopted. For one, each actor within the ecosystem has its own way of working that can hardly be standardized [Bosch, 2009]. This results in many different practices applied across the ecosystem and, therefore, many different degrees of agility which can be difficult to align. Moreover, the actors do not belong to a single but instead to many different organizations which share different, possibly competing, relationships. For this reason, the actors do often neither share the same goals nor communicate in a fully open way. This makes the inter-team coordination even more difficult.

To our knowledge, the challenges of inter-team coordination beyond organizational boundaries are not yet sufficiently investigated in existing literature. For that reason, we raise the following research questions:

a) How do agile teams within software ecosystems coordinate their efforts?

b) Which inter-team coordination challenges do agile teams within software ecosystems face?

In order to achieve this, we conducted a case study within two large, industrial software ecosystems to investigate their processes and inter-team coordination. We elab-
orrate the results along three dimensions that constitute the framing for our findings: a) maturity of the ecosystem b) phases within the agile lifecycle c) openness / closedness of the ecosystem. We use this framing to map the conflicting interests between actors as well as the resulting challenges and implications to the dimensions. Therefore, the contribution of this paper is to unfold why certain conflicts arise in a particular situation or setting in order to increase awareness of other actors' mindsets, and to help practitioners understand certain challenges and possible trade-offs they, thereby, might be facing. Moreover, we were able to tie most of the challenges back to long communication paths and a lack of established processes, raising the need to extend existing agile practices.

The remainder of this paper is organized as follows: First, we explain the characteristics of software ecosystems in Section 2.2, followed by our case study design in Section 2.3. In Section 2.4 we present the results of our study, before providing an overview of related work in Section 2.5, and summing up and concluding our work in Section 2.6.

### 2.2 Characteristics of Software Ecosystems

One of the major differences between distributed teams in traditional organizations and software ecosystems is the fact that the teams or actors are not within the same company or organization but instead spread across several organizations, whereby each actor contributes different elements to the system or product. This entails various types of relationships between actors within an ecosystem. For instance, the actors can be competitors or share mutual benefits [Manikas and Hansen, 2013]. The complexity of relationships and dependencies increases with the number of parties involved in the ecosystem [Bosch, 2009]. Moreover, the number of actors and their possibly competing relationships results in a lack of sharing data which has previously been observed in large organizations [Fabijan et al., 2016].

Even though iterative requirements engineering processes provide an increased flexibility and are already widely applied in software ecosystems, further challenges arise due to the ecosystem's various actors, the physical distance between them, and the complexity of dependencies within the ecosystem, which impede a common understanding among and alignment between actors [Fricker, 2009], [Knauss et al., 2014], [Valença et al., 2014]. For one, the interpretation and prioritization of requirements can differ highly between ecosystem actors because different stakeholders value different attributes when dealing with a requirement. Every partner contributes requirements to the ecosystem which might result in a requirement overload, causing complications in the prioritization process [Karlsson et al., 2002]. Additionally, the negotiation process of requirements is highly influenced by the amount of power and dependencies between actors [Valença et al., 2014]. An adequate understanding of the other actors' goals and business models is required in the requirements engineering process in order to satisfy existing stakeholders and attract new partners.

In this paper we analyze how the described characteristics and challenges manifest in the inter-team coordination of agile teams within software ecosystems.
2.3 Research Design

Case studies are a well-known research methodology to investigate and understand contemporary phenomena in their real-world context with no or little control by the researcher [Runeson and Höst, 2009], [Yin, 2014]. As our research questions aim at answering exploratory questions, we believe that this is the right methodology for our study, following the guidelines by Runeson and Höst [Runeson and Höst, 2009].

2.3.1 Case study design

The research objective of our study was to investigate the inter-team coordination and its accompanying challenges across organization boundaries. Specifically, our study focuses on how teams in distributed organizations communicate with each other, what kind of information, feedback or data they share, how it is shared, and how they are making and communicating decisions that affect any of the other teams. To achieve this, we performed this case study in two large software ecosystems, Ecosystem A and Ecosystem B, which are established in the industrial and the healthcare domain, respectively. Each ecosystem originated in a large, industrial company and offers several services, mostly in terms of applications, to the companies’ customers. In order to expand their offerings, they opened up their platform to internal as well as external partners developing applications on the platforms. Figure 2-1 gives an overview of the ecosystems’ structures and actors. We chose the respective ecosystems for our study since the keystone as well as the partners work in agile teams and experience difficulties in their coordination.

![Figure 2-1: Actors in software ecosystems](image)

<table>
<thead>
<tr>
<th>Ecosystem</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td># of platform devs</td>
<td>500-750</td>
<td>50-100</td>
</tr>
<tr>
<td># of internal partners</td>
<td>20-50</td>
<td>5-10</td>
</tr>
<tr>
<td># of external partners</td>
<td>100-200</td>
<td>5-10</td>
</tr>
<tr>
<td># of apps</td>
<td>20-50</td>
<td>10-20</td>
</tr>
<tr>
<td>Interviews Keystone</td>
<td>DM</td>
<td>PO</td>
</tr>
<tr>
<td>Interviews Partner</td>
<td>SA</td>
<td>PO I</td>
</tr>
</tbody>
</table>

Table 2.1: Description of ecosystems
2.3.2 Data collection & analysis

We conducted semi-structured interviews with key stakeholders within the two ecosystems. Four product owners (PO), one product manager (PM), one demand manager (DM), and one software architect (SA) participated in the study. The demand manager is responsible for collecting and structuring requests from partners and customers before forwarding them to appropriate platform teams. Each of our interviewees belongs to a different, individual agile team within the respective ecosystem and was chosen as a key stakeholder to represent the views of their entire team. Moreover, all interviewees belonged to either the keystone who develops the platform (PM, DM, and one PO) or to a complementing player (three POs, SA) of an ecosystem, therefore providing views from both angles. One product owner, the demand manager, and the software architect belonged to Ecosystem A and the product manager and three of the product owners belonged to Ecosystem B (see Table 2.1 for a structured overview of the ecosystems and our participants).

All interviews included the following topics: communication structures, exchange of data and feedback with partners and customers, and decision making processes; though the interview guides were slightly adjusted to the specific roles. At the beginning of each interview the participants were given a brief introduction into the study and the structure of the respective ecosystem was shortly discussed in order to create a common understanding. Following this, the interviewees were asked for their permission to audio tape the interview, before the interviews were conducted. Each interview lasted between 45 minutes and one hour, and was transcribed and summarized afterwards.

Additionally to the interviews we derived further knowledge from two experts in this field. Both of them have been working in and with multiple ecosystems, including Ecosystem A and Ecosystem B, for several years, and shared their experiences in a couple of unstructured interview sessions. They provided additional insights on how the respective ecosystems are structured from a bird's-eye perspective in contrast to the team perspectives. Moreover, we discussed the findings of our interviews with them in order to support the validity of our study.

As a result, we achieve triangulation by a) investigating multiple software ecosystems b) interviewing multiple roles within the ecosystems c) adding expert knowledge.

2.4 Results

One major difference between distributed teams in large organizations and teams within software ecosystems is that the actors within an ecosystem do not belong to the same company or organization and, therefore, do not necessarily share a common (business) goal. Since we wanted to understand why certain challenges occurred, we decided to first investigate the respective (differing) interests on the keystone’s as well as the partners’ side in order to locate potential sources of conflicts, before we focused on the analysis of the challenges. Hence, this section is structured as follows: First, we describe the dimensions used in our framework, followed by the conflicting
interests, and concluding with the identified challenges.

2.4.1 Influencing Factors

During the analysis of our data we observed that our findings were highly dependent on three different factors: the phases of an agile lifecycle comprise different tasks that require different communication and coordination processes, and the maturity as well as the openness of an ecosystem influence the relationships between actors and, therefore, also the communication. These factors constitute the main dimensions of the models that include the results of our study, and are explained in detail in the following sections.

Agile Lifecycle

The common agile lifecycle includes an initial requirements definition and planning phase, followed by a development phase including integration and tests, a review and feedback phase, frequent releases, and a reprioritization phase before the next iteration begins [Burger, 2016]. Since all our interviewees work in agile teams, they all go through a similar agile lifecycle, experiencing different challenges in different phases. As the focus of our study is on coordination challenges, we neglected the technical phases (development and release) but rather focused on the planning, prioritization and feedback phase. We asked each interviewee about the phases they go through and, based on the literature as well as their answers, we propose the agile lifecycle in Figure 2-2 that constitutes one dimension in our study.

![Figure 2-2: Agile lifecycle](image)

Maturity

Based on our interviews, we noticed that the maturity of the respective ecosystems plays a rather important role. Especially the communication between keystone and partners changes drastically over different phases of maturity. For instance, in the early phases of opening up a platform it is important to attract new and please existing
partners, therefore the focus on communication is much higher than in later phases when the ecosystem is mature enough to attract partners automatically because of its success and the benefits for the partners.

One way to describe the evolution of a technology or an innovation is the s-curve. It describes the performance of a technology during different maturity stages from "pregnancy, birth, childhood, adolescence, maturity, and decline" [Slocum, 1999]. Both ecosystems already have products on the market but regarding their maturity we would classify Ecosystem A as still being in the "birth" phase and Ecosystem B as being in the "childhood" phase. Specifically, Ecosystem A is still in the process of opening up their development to external partners while Ecosystem B is already established but still accelerating. For this reason, we define the second dimension as the maturity from "opening up" to "acceleration".

Openness vs. Closedness

Hartman et al. identified two different types of ecosystems, open and closed [Hartmann et al., 2012]. While closed ecosystems are still tightly coupled to and somewhat controlled by the keystone, open ecosystems are easily accessible for partners and can be characterized by their interchangeability of components and parties. However, ecosystems are not necessarily one or the other, they can also be in a hybrid stage [Hartmann et al., 2012]. This is relevant for our study since the communication and trust across ecosystem partners appears to be different for the two types. For instance, the actors in closed ecosystems are more interconnected than the actors of open ecosystems and, therefore, communicate more openly and share a higher level of trust. By tendency, Ecosystem A belongs to the category "open ecosystem" while Ecosystem B incorporates closed as well as open aspects. This is directly reflected in the characteristics of relationships that the keystone shares with different types of partners (both external & internal).

2.4.2 Conflicting Interests

Each ecosystem actor usually follows its own business strategy which can easily result in different interests that are hard to align and, therefore, cause conflicts between the actors. In order to analyze which opposing interests might lead to conflicts, and based on that even challenges, we extracted the interests of the partners as well as the keystone out of the interviews. Next, we mapped contrary interests that share a common link from both sides to each other which, therefore, constitute main drivers and crucial influencing factors for certain conflicts. Since not all interests and conflicts apply to all ecosystems or to all phases of agile development, we mapped the conflicting interests to different maturity levels, phases within an agile lifecycle, and the degree of openness (see Figure 2-3). A more detailed description of the conflicts, interests of the platform and partners, and other influencing factors can be found in Table 2.2. All of the described results were derived out of the interview sessions of our case study. Overall, we identified nine conflicts that were caused by opposing interests on the partners' and the keystone's side. Three of the conflicts originated
in the planning phase, four in the value prioritization phase, and two in the feedback phase.

Planning Phase

Conflict #1 concerns the platform functionalities provided by the keystone and required by the partners. On the one side, the partners require the keystone to provide specific functionalities for their minimum viable product (MVP) while the keystone receives so many requests that it is difficult to take all partners into consideration so "[they] need to come to a point where [they] say what platform value creates the most value to the ecosystem and that’s not easy because the ecosystem is so broad". For this reason, the keystone concentrates its planning efforts on the partners that bring the most value to the ecosystem.

The next issue (#2) is related to the communication of requests and the inconsistency of processes. Partners want to communicate their requests in an easy and fast way, and expect the keystone to respond to their requests, while the keystone "receive[s] quite small or tiny, tiny described requests" but "would like to receive more well described requests" from their partners. As there are no consistent processes available this often leads to misunderstandings of what the requirement actually is and, therefore, causes displeasure on both sides.

Moreover, the keystone’s control and the partners’ independence (#3) lead to conflicts, especially in closed ecosystems. While the partners want to be independent of the keystone in order to being able to optimize individual business interests, the keystone wants to keep control over the end-customers and the partners’ interactions with these customers in order to ensure a coherent, overall business offering.
<table>
<thead>
<tr>
<th>#</th>
<th>Conflict</th>
<th>Partners' Interests</th>
<th>Keystone's Interests</th>
<th>Ph</th>
<th>Ma</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Platform functionalities</td>
<td>Request specific partner functionalities</td>
<td>Provides the functionality that brings most value to the ecosystem</td>
<td>P</td>
<td>OP</td>
<td>O</td>
</tr>
<tr>
<td>2</td>
<td>Communication of requests</td>
<td>Expect fast &amp; easy communication processes</td>
<td>Asks for well described requests</td>
<td>P</td>
<td>OP&amp;A</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>Keystone’s control &amp; partners’ independence (planning)</td>
<td>Become more independent</td>
<td>Keep control over the partners’ customer interaction</td>
<td>P</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>Prioritizations &amp; different business strategies</td>
<td>Follow own business strategy</td>
<td>Ensure the ecosystem’s future</td>
<td>VP</td>
<td>OP&amp;A</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Different power relations by the keystone</td>
<td>Want to be recognized by keystone</td>
<td>Wants to bind &quot;important&quot; partners to ecosystem</td>
<td>VP</td>
<td>OP</td>
<td>PR</td>
</tr>
<tr>
<td>6</td>
<td>Transparency by the keystone</td>
<td>Expect transparency (e.g., delivery timelines, commitments)</td>
<td>Wants to stay flexible / be able to reprioritize</td>
<td>VP</td>
<td>OP</td>
<td>O</td>
</tr>
<tr>
<td>7</td>
<td>Keystone’s control &amp; partners’ independence (prioritization)</td>
<td>Obtain broad picture over all customers</td>
<td>Keep control over the partners’ customer interaction</td>
<td>VP</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>8</td>
<td>Exchange of data &amp; required infrastructure</td>
<td>Share customer feedback/data with collaborative partners</td>
<td>Low priority to provide infrastructure</td>
<td>F</td>
<td>OP&amp;A</td>
<td>O</td>
</tr>
<tr>
<td>9</td>
<td>Forwarding customer feedback</td>
<td>Only benefit from forwarding feedback if it is directly connected to the partner’s app/service</td>
<td>Needs the partners to forward requirements (if related to the keystone) of their customers</td>
<td>F</td>
<td>OP&amp;A</td>
<td>O</td>
</tr>
</tbody>
</table>

Phases (Ph): Planning (P), Value Prioritization (VP), Feedback (F)  
Maturities (Ma): Opening up (OP), Acceleration (A)  
Influencing factors (IF): Openness (O), Closedness (C), Power Relations (PR)

Table 2.2: Description of conflicting interests between keystone and partners and the influencing factors in different phases

Value Prioritization Phase

Conflict #4 concerns *prioritizations and different business strategies*. As each partner follows its own business strategy, the keystone wants to ensure the ecosystem’s future which leads to conflicts in the prioritization process since the different strategies can be difficult to align. One interviewee explains that they need to decide "from a platform point of view, what makes most sense, what is scalable, what is beneficial for a lot of customers [...] that’s a challenge".

It is quite natural that some ecosystem partners are more important business partners to the keystone than others. However, this leads to *different power relations* (#5) as the important partners can create more pressure on the keystone than the others. Ultimately, the partners expect the keystone to be aware of them and their needs and to be treated (at least) equally to other partners, while the keystone wants to bind its important partners (e.g. defined by the number of customers, revenue etc.) to the platform.

Moreover, especially during that opening up phase, the ecosystem partners expect transparency of the keystone, e.g. concerning the prioritization of next steps, delivery timelines, or commitments. One of the interviewees states that he "would like to have more transparency how and on what basis decisions are made [...] because at the moment it’s very non-transparent how [the keystone] decides what constitutes
the biggest value for the overall project". However, the keystone avoids giving too detailed commitments and detailed timelines in order to stay flexible and being able to reprioritize. This leads to conflicting interests concerning the amount of transparency by the keystone (#6).

Analogously to and building upon conflict #3, the keystone’s control and the partners independence (#7) create a conflict in the prioritization phase of closed software ecosystems. The partners would like to base their prioritization on a broad picture of all their customers while the keystone wants to keep control over the interaction with customers.

**Feedback Phase**

The partners would like to receive as much information on their customers as possible even if the data is collected by another partner. However, they are disinclined to share their data with competitors within the ecosystem but would be willing to do so with collaborative partners. On the other hand, the keystone perceives it as low priority to provide an infrastructure for sharing data across the ecosystem. This leads to conflicts concerning the exchange of data and the required infrastructure (#8) to do so.

Lastly, and related to the previous conflict, the handling and forwarding of customer feedback (#9) concerning other partners or the keystone also constitutes certain challenges since the partners only benefit from forwarding feedback if it is directly connected to the partners’ apps or services. However, the keystone relies on the partners to forward platform-related requirements of their customers to them. Additionally, one interviewee explains that "feedback from customer visits are a tricky thing because the POs are going to the visits and we have a process described on how to feed back the feedback to the organization and also to me but that is still a little bit... some use it, some don’t and some you have to chase to get their feedback for their customer" which is why they "need to turn that into a more automated, also tool-automated, process flow".

**2.4.3 Challenges**

Based on the previously extracted conflicting interests, all ecosystem related challenges faced by either the keystone or the partners were extracted out of the interviews. For each challenge we identified the following properties: The ecosystem’s maturity, the phase within the agile lifecycle, other influencing factors, and causes for the respective challenge. Table 2.3 shows an overview of the detected challenges. In a next step, we mapped the challenges into a multi-dimensional model (see Figure 2-4). The two main dimensions are the phases within the agile lifecycle (planning, value prioritization, and feedback) and the degree of maturity of the ecosystems. We added an extra dimension, the openness of the ecosystem, to each of the phases of the agile lifecycle because we identified challenges within these phases that were also strongly influenced by this factor. We identified two types of partners: partners that are closely coupled to or guided by the keystone and partners that are only loosely
coupled to the keystone. Challenges between actors may be effective only in one or in both directions (arrows with one vs. two heads in Figure 2-4). The power balance can be even or be dominated by one actor (indicated by a square or by triangles respectively).

Planning Phase

The first challenge results out of conflict #1 concerning the development of basis or new platform functionalities. The partners sometimes feel like the keystone is not sufficiently responsive and takes too long to deliver needed functionalities while the keystone receives too many requests over a lot of different channels which makes it difficult to respond to or handle the requests in a decent amount of time, as one interviewee explains "we keep on getting requests from everywhere [. . .] not everything can be taken up at the same time". The challenge is to achieve the right request responsivity (#1). Among others, this is caused by missing or not well established processes to handle such requests which, again, leads to many different communication channels.

Furthermore, both cases in our study perceived an appropriate communication of topics and deliverables (#2) between platform and partners as quite difficult as a result to conflict #2. The partners reveal that they would appreciate more transparency on the keystone’s side in order to get a clear picture of what is possible to achieve. If this is not well communicated, this lack of transparency easily leads to a lack of trust. On the other hand, the keystone explains that they mostly receive high level user stories from their partners which have a high potential for being misinter-
<table>
<thead>
<tr>
<th>#</th>
<th>Challenge</th>
<th>Description</th>
<th>Ph</th>
<th>Ma</th>
<th>IF</th>
<th>PR</th>
</tr>
</thead>
</table>
| 1  | Achieve sufficient request responsivity                                   | P: Keystone not sufficiently reactive  
   K: Keystone receives too many requests over a lot of different channels  
   Why: many different communication channels, processes not well established yet                                                     | P   | O   | O   | P>K |
| 2  | Appropriate communication of topics & deliverables                         | P: Lack of transparency & lack of support - leads to lack of trust  
   K: Apps formulate high level user stories - Keystone PM refines it  
   - potential misunderstandings  
   Why: long communication paths, not all PMs have expertise in all areas - easy to misunderstand                                              | P   | O&P | O   | P>K |
| 3  | Obtain a broad picture of customers (planning)                              | P: Partners get restrictions from keystone concerning customer interaction - no broad overview on customer's needs  
   K: Keystone wants to ensure an appropriate representation of the entire ecosystem in front of the customer  
   Why: partners & keystone closely coupled, keystone does not want the customer to see the product in a non-ready state               | P   | A   | C   | K>P |
| 4  | Achieve alignment of roadmaps and prioritizations                          | P: Every ecosystem partner has own roadmap and prioritizations do often not match  
   K: Challenging to consider all partners & decide what brings most value to ecosystem  
   Why: no common business interests in software ecosystems                                                                                                                                         | VP  | O   | A   | PR  | P>K |
| 5  | Handling different power relations                                         | P: Keystone gives preference to certain stakeholders - neglect "less important" partners  
   Why: Not all partners can be treated the same  
   K: Partners create pressure in order to get their requests preferred - how to decide which partner/customer is more important?  
   Why: Keystone relies on "powerful" partners in opening up phase                                                                             | VP  | O&P | PR  | P>K |
| 6  | Insufficient communication of prioritizations                              | P: Prioritizations are not well communicated  
   K: Challenge to handle tradeoff between pleasing partners and maintaining flexibility  
   Why: Keystone wants to stay flexible                                                                                                      | VP  | O   | O   | P>K |
| 7  | Obtain a broad picture of customers (prioritization)                        | P: Limited communication between customers and partners - not all customers included in prioritization process  
   K: See challenge #3  
   Why: as a result of #3                                                                                                                | VP  | A   | C   | K>P |
| 8  | Establish processes to collect & exchange data                              | Partner & Keystone: No exchange of customer feedback within ecosystem, no direct Feedback channel / centralized way to store & access feedback - information loss, "one-sided" feedback  
   Why: off-the-shelf infrastructure can usually not be used, do both partners/keystone benefit by sharing?                               | F   | O&P | O   | -   |
| 9  | Appropriate communication & avoidance of information loss/altering         | P: Insufficient communication of malfunctions by the keystone  
   Why: communication channels for incident reporting must be established, more challenging in ecosystem  
   K: Limited amount of information & information loss when communicating across multiple ecosystem partners  
   Why: Multiple alterations due to long communication paths, feedback from end-customers communicated via partners                   | F   | O&P | O   | -   |

Table 2.3: Description of coordination challenges caused by divergent interests of actors within the ecosystem
interpreted by the product manager who has to refine the user stories. Possible causes for this challenge are long communication paths across multiple stakeholders often implying information loss, and the fact that it is impossible for product managers to have expertise in all of the partners’ areas which easily leads to misunderstandings, especially since the partner offering are "operating in a very specific domain which makes it difficult for most people to understand the topics".

Another challenge, closely coupled to closed ecosystems and related to conflict #3, is to obtain a broad picture of the end-customers (#3) due to keystone guidelines. In case partners and the keystone are very tightly coupled, partners who talk to customers are perceived as representatives of the entire ecosystem. For this reason, the platform wants to ensure an congruent representation of the ecosystem in front of the customers. In the particular case, the keystone has collaboration agreements with certain customers and the partners get instructions which partners they should talk to. However, this keeps the partners from getting a broad overview over all customers.

**Value Prioritization Phase**

As a result to conflict #4, it is challenging to achieve alignment of roadmaps and prioritizations (#4) as the actors within an ecosystem simply do not share a common business interest. This makes it difficult for the keystone to consider all partners and decide what brings the most value to the ecosystem. One of the partners states that "it is challenging because the keystone has its own roadmap, its own prioritizations and this can cause conflicts if the priorities or values do not match".

Quite the contrary to the previous challenge and related to conflict #5, this challenge addresses the difficulty of handling different power relations (#5) within the ecosystem. The partners feel like the keystone gives preferences to certain "important" customers and neglects the "less important" customers. One interviewee explains that he feels like "it depends on which partner has the greatest business potential". However, the keystone is simply not able to treat all partners with the same amount of attention because "[the partners] are always creating pressure" in order to get their requests preferred which naturally leads to the questions how to decide which partners are more important.

Conflict #6 concerning the amount of transparency by the keystone leads to an insufficient communication of prioritizations (#6), e.g. concerning the keystone’s next steps, which causes displeasure on the partners’ side. At the same time, the keystone faces the challenge how to handle the trade-off between pleasing partners and maintaining its flexibility.

As a result of challenge #3 and related to conflict #7, we observed that – due to the divergent viewpoints concerning the partners independence and the keystone’s control – the partners face the challenge of obtaining a broad picture over all their customers (#7). The reason is that they are unable to include all their customers in their prioritization process due to the keystone’s limitations concerning the customer communication. One of the interviewees even states that "It would be much better to run more statistics because I don’t feel like this is a comprehensive picture".
Feedback Phase

Some of the interviewees reported on their interest in collecting and sharing customer data with certain collaborative stakeholders (conflict #8), however, so far there exist no established processes for software ecosystems to do so. The interviewees explain that they "would rather have direct feedback channels" or "centralized ways to store and access feedback" because off-the-shelf infrastructure can usually not be used for such kind of data sharing since fine-grained access to the data is not easy to control and legal or privacy issues need to be addressed. As a result, this leads to information loss and one-sided feedback. Therefore, the challenge is to establish processes to collect and exchange data (#8).

Lastly, both of our cases perceive the communication across stakeholders as insufficient and are under the impression that a lot of information gets altered or lost due to the (mis-)communication across multiple partners. This results in the challenge of an appropriate communication and avoidance of information loss and altering (#9). One of the interviewees revealed that the keystone does not immediately communicate malfunctioning platform features that the partners’ features rely on to them because high priority communication channels for incident reporting in software ecosystems would need to be established first. On the other hand, the keystone suffers from limited amount of customer feedback and information loss since the chances for alterations are very high due to the long communication paths. One interviewee reports that "the more people you involve in between the more information gets lost". Moreover, the feedback from end-customers concerning platform features are often communicated via the partners. Especially in open ecosystems the keystone often does not have direct customer contact. This makes it challenging for the keystone to receive information on and to understand the real customer’s needs.

2.5 Related Work

Previous research suggests that the application of agile practices is more difficult in large projects or organizations than in small teams [Dybå and Dingsøyr, 2008]. As an organization grows it becomes challenging to keep an overview of all projects and groups within one organization [Stettina and Schoemaker, 2018]. Additionally, if the activities between them are not well communicated, it is hard to keep track of the existing dependencies. These factors often result in coordination challenges and additional coordination efforts [Bick et al., 2018], [Dikert et al., 2016]. For instance, an overarching figure or role, as well as appropriate methods, are required to coordinate the teams and address team-crossing challenges [Uludağ et al., 2018]. However, inter-team coordination in software ecosystems rises additional complications as the teams are distributed over several organizations who rarely share common goals or strategies nor a centralized control figure who coordinates them.

Moreover, it has been observed that an increased autonomy enabled by agile practices causes individual teams within a mutual organization to prioritize their own goals over the larger context [Dikert et al., 2016]. Knowledge regarding the system
is spread across the distributed teams and processes to share that knowledge need to be established [Rolland, 2016], [Moe et al., 2016]. Additionally, a global distribution of teams leads, among other challenges, to "reduced feelings of proximity when telecommunication is necessary, and difficulty in arranging frequent meetings due to time zone differences" [Dikert et al., 2016].

Previous studies by Dingsøyr et al. [Dingsøyr et al., 2018] and Stettina et al. [Stettina and Hörz, 2015] indicate that most issues identified in agile large-scale software projects are related to processes as well as the people and their relationships. We observed quite similar results in our case study. Nevertheless, our results differ from the challenges of distributed agile teams in the way that the interactions between teams of a single organization are quite different to the interactions across organizations. In the latter case, single parties do not necessarily share a mutual larger context or even a common business interest which also impedes the sharing of knowledge. Moreover, the individual teams do neither apply or utilize unified processes nor can they be forced to do so since a central control figure does not exist in this context.

In order to improve the lack of visibility in large-scale projects, methods and solutions such as agile portfolio management, reporting or inter-team retrospectives have been introduced to connect the business strategy and the respective teams, to get an overview of all initiatives within a portfolio, and to address inter-team coordination challenges [Rautiainen et al., 2011], [Dingsøyr et al., 2018]. However, it has not been investigated yet how such practices could be applied across organizational boundaries. For instance, some actors can be reluctant to share their reports with certain other actors. Therefore, practices or guidelines would need to be established to coordinate the distribution of reports or to enable inter-organization retrospectives.

The complexity of (inter-)team coordination tends to increase with the size of the project and the number of teams involved (e.g. in multiteam systems). A shared mental model (e.g. concerning the work process, tasks, or awareness of who knows what), closed-loop communication, and trust are considered mechanisms that facilitate the coordination of multiple teams. Bjørnson et al. [Bjørnson et al., 2018] investigated the practices that can be applied in order to implement these mechanisms. They identified, among others, formal as well as informal communication channels, specialized roles that rotate between teams, stand-up meetings, mini demos, and discussions in an open workspace as helpful tools to implement the mechanisms. In their case study, all teams are located in the same office and many of the proposed practices rely on the co-location of the teams, or at least on a shared common business interest and willingness to exchange information. These characteristics can usually not be observed in software ecosystems which makes it challenging to adapt these practices and mechanisms in this context.

Scheerer et al. [Scheerer et al., 2014] investigated different types of coordination strategies for multiteam systems. Each of their strategy types comprises three coordination types – mechanic (e.g. plans, rules), organic (e.g. mutual adjustment, feedback), and cognitive (e.g. cognitive similarity configurations) – that are applied to different kinds of extents, e.g. low, medium, or high. This work specifically focuses on multiteams that work on the same software product and while each team works toward an individual goal, they still share, at least to some extent, a mutual collective
goal [Scheer et al., 2014]. Rolling out a unified coordination strategy ecosystem-wide is very difficult to enforce since it would affect teams across several organizations, each of them pursuing their own goals and applying their own practices, without sharing a central control figure.

2.6 Conclusion

The research objective of our study was to elaborate the arising coordination challenges of agile teams within software ecosystems. Our findings indicate that many of the identified coordination challenges are either directly or indirectly related to long communication paths and a lack of well established communication processes, especially if information needs to be shared with other actors across organization boundaries. In contrast to distributed teams within one company, this is additionally challenging because of the varying, sometimes even competitive, relationships that influence the communication and the way data is forwarded or shared. For one, our participants perceived the responsivity as very slow and insufficient. Moreover, the deficient communication structures cause a lack of awareness and understanding of topics, deliverables and timelines between the keystone and its partners. The keystone is rather cautious when it comes to revealing its prioritizations and plans for the future which causes frustration on the partners’ side. In addition to that, our results imply that on many occasions information gets lost or altered due to the multiple hops it has to pass. Our research provides evidence that there is a need to adapt or develop agile processes to facilitate and enable across-organization communication, coordination, and exchange of data. Therefore, future work could be dedicated to solving the identified challenges and to investigate how agile practices would need to be adapted in order to fit across-organizational needs.
Chapter 3

Business as Unusual: A Model for Continuous Real-time Business Insights Based on Low Level Metrics


Abstract

A wide variety of tools to monitor and track software systems, such as websites or smartphone applications, during runtime already exists. However, their aggregated results are often not sufficient to answer questions on a product management level since these questions address several levels of complexity and abstractions, and tend to be formulated on a rather high level, for instance concerning the efficiency of their website structure for their users. A straightforward mapping between low level metrics and high level insights is typically not possible. This causes a gap that makes it challenging to continuously provide quantitative high-level insights in real-time. In order to address this challenge, we conducted a study within three distinct platforms and products, and propose a model based on our results. After defining a case for each of the independent platforms and products, we implemented a process to measure high level insights using low level metrics for each of these cases. Next, we compared the procedures and steps that were taken in each of the cases and derived a model that describes a generic approach how to utilize and process data in order to gain higher level insights. Our model structures the steps from data to knowledge over different levels of complexity and abstraction, namely operational, tactical, and strategic. Thereby, the knowledge acquired in each phase serves as input in the next phase which increases the measurable level of complexity with each iteration. Since the steps in our model are specifically arranged as a pipeline, it enables practitioners to automate a continuous and quantitative measurement of high level insights in real-time.
3.1 Introduction

Even though they think they know what their customers want, producers often make incorrect assumptions on what actually constitutes value to their customers [Butz Jr. and Goodstein, 1996]. It is crucial to understand how a product is used by its customers and how to utilize these insights and turn them into actions. However, it can be difficult and time-consuming to gain a broad understanding of all customers since each customer may value different aspects of the product [Butz Jr. and Goodstein, 1996]. Determining the value of a feature has proven to be a difficult task, especially since the respective values can evolve dynamically over time. In order to identify features that lack or have lost their value, it becomes important to monitor the customers’ perceptions of the features in order to adapt product strategies. Currently, most of the existing approaches are based on interviews, and therefore subjective opinions [Marciuska et al., 2014]. In the age of DevOps and continuous deployment, customer behavior can change just as fast. This makes it difficult to track the customers’ perceptions of the product through qualitative studies in a continuous way, therefore raising the need for real-time quantitative measurement of high level business indicators. This, however, is not a trivial task. On the one side, companies are still struggling to make actual and meaningful use of the data [Bizer et al., 2012], [Olsson and Bosch, 2013]. It is difficult to keep track "what is collected and for what purpose" [Olsson and Bosch, 2013]. Additionally, insights can also be spread across multiple sites, therefore being stuck in isolated silos. Companies often struggle to establish a systematic sharing process within their organizations [Fabijan et al., 2016]. On the other side, a simple one-to-one mapping between high level insights and the collected quantitative data is usually not feasible. Rather, a whole range of methods needs to be applied in order to reach a certain degree of complexity, accuracy and trustworthiness to describe the required high level insights. This fact significantly impedes the described requirements of automating such a process for real-time measurements.

In order to address this, we conducted a study within three different platforms and products, and propose a model based on our results. Our goal was to close the gap between low level metrics and high level insights, and to enable an automated real-time measurement for these insights based on quantitative data. We implemented this process for three different and independent cases (one for each platform or product). Next, we derived a generic, reusable model from the individual procedures, consisting of several iterations that we went through in each case. Since the types of insights and their corresponding techniques evolve with each iteration, we mapped the iterations to three different levels, namely operational, tactical, and strategic. These levels constitute one dimension of our model while the process of turning data into high level insights constitutes the second dimension.

The contribution of this paper is twofold. First, we provide a structured approach how to systematically implement a process to measure high level business insights by using multiple blocks that build upon each other, and therefore incrementally lifting up the insights to a business level perspective. Second, the steps in our model are specifically arranged as a pipeline which enables practitioners to automate this
process as a continuous feedback loop and, therefore, to quantitatively measure high level insights in real-time.

The remainder of this paper is structured as follows: First, we present the background and motivation of our study in Section 3.2, followed by the research method used in our study in Section 3.3 and the implementation details in Section 3.4. The results are presented in Section 3.5, while Section 3.6 gives an overview of the related work on this topic. Finally, we conclude and summarize our work in Section 3.7.

3.2 Background and Motivation

Over a period of five months, we conducted a study within a large-scale software ecosystem operating in the healthcare domain. Our intention was to understand how the teams utilize the data produced during the runtime of their product and where they struggle to make actual use of the data. The ecosystem itself is divided into a platform as well as several applications that are hosted on the platform. During the course of our study we interviewed six product owners of different areas, including multiple applications, the platform itself as well as operations. Following this, we set up a collaboration phase. While we started to run some first analyses on their data, this phase also consisted of multiple knowledge exchange sessions, unstructured interviews, and working / brainstorming sessions, taking place about once a week over a period of four months.

The results of our interviews indicate that on the one hand, the product owners describe their needs on a rather high level and struggle to specify which measurements constitute a value to them, while the operations team experiences the gap between having large amounts of data and many low level metrics that they can measure and the high level requests they receive from the product owners. These results as well as the observation that other platform and product teams tend to experience similar challenges, served as a motivation for us to set up this study.

3.3 Research Method

We chose an inductive study design in order to derive our model from actual real-world cases [Goddard and Melville, 2004]. We selected three different and independent cases from three different platforms or products, and developed a process to quantitatively measure high level business insights for each case individually. We documented and compared the different procedures in order to derive a generic model. The following subsections will give a more detailed overview of the selected cases.

3.3.1 Case A

For the first case of our study we collaborated with one of the product owners as well as the operation team of the software ecosystem described in Section 3.2. We conducted multiple unstructured interviews in which we identified the kind of insights
they would like to receive. Specifically, they would like to know how a certain application is used by its customers. In more detail, they would like to understand how certain levels within the website structure are accessed, what they can learn from the behavioral usage patterns, and where the application could potentially be optimized regarding usability. In summary, the high level insights generated in this case are a representation of the current usage, an understandable description of usage patterns as well as recommendations concerning structure optimizations based on the usage patterns.

3.3.2 Case B

For the second case of our study we worked with the software architect of a platform forming the basis of a software ecosystem and operating in the industrial domain. Several applications run on the platform and are developed by separate business units and different divisions of the company (= ecosystem partners). The platform’s product management is uncertain how their platform is used by the application developers. Case B therefore targets the questions what platform APIs are used, how they are used, and how the provider can improve its platform offerings for its ecosystem partners. In particular, the high level insights generated in this case are a representation of the current API usage, an understandable description of the relationships within the API usage as well as recommendations how the platform can improve their offering in order to simplify the API usage for their partners.

3.3.3 Case C

For the last case of our study we collaborated with a product manager and software architect of a platform that provides services to establish remote connections to industrial devices and plants. It includes a user management that allows the assignment of different roles and rights to users. They document all operations performed on their system in audit logs. The questions defined in alignment with the product management are related to the most frequently executed operations as well as common operation patterns. In more detail, the high level insights of this case are, similarly to the previous cases, a representation of the current operation executions as well as a description of common operation sequences, and recommendations to condense certain sequences in overall operations in order to reduce the number of operations needed to perform certain tasks.

3.4 Implementation

The following subsections will provide an overview of the procedures we have implemented for Cases A-C.
3.4.1 Case A

The procedure of Case A is described in the following subsections. Additionally, Figure 3-1 provides a first overview of the step-wise approach.

![Diagram of Case A Procedure]

**Step I**

The input data of step I comprises many different and unconnected low level metrics. In the beginning, we extracted the relevant data sets that are related to the questions derived from the product owner's interests, and performed basic aggregation techniques to provide a certain understanding of the data and some basic insights. This serves as the groundwork that all further steps are built upon. These aggregations and their respective visualizations aim at answering basic questions on a rather operational level. For this specific case we extracted all metrics related to the application usage: number of page views, maximum, minimum and average duration on each page, number of navigation paths, length of navigation paths etc. Next, we examined how often users visited a certain page in comparison to other pages, how much time users spent on each page (see Figure 3-2), the relationship between pages (e.g. how many times the user navigated from one page to another, see table 3.1), and how often certain functionalities were used. As a result, the product owner can derive the basic navigation paths users usually take while using the application. Additionally, by knowing the frequency of the page views on different levels within the website structure as well as the duration the user spent on each of the pages, the product owner is able to determine how important certain pages or features are for the customer.
Table 3.1: Most common navigation paths

<table>
<thead>
<tr>
<th>Source Page</th>
<th>Target Page</th>
<th>Path Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>appname/overview/</td>
<td>appname/customers/</td>
<td>1286</td>
</tr>
<tr>
<td>appname/overview/</td>
<td>appname/devices/</td>
<td>1177</td>
</tr>
<tr>
<td>appname/overview/</td>
<td>appname/actions/</td>
<td>870</td>
</tr>
<tr>
<td>appname/utilization/</td>
<td>appname/devices/</td>
<td>785</td>
</tr>
<tr>
<td>appname/customers/</td>
<td>appname/actions/</td>
<td>717</td>
</tr>
<tr>
<td>appname/actions/</td>
<td>appname/utilization/</td>
<td>619</td>
</tr>
<tr>
<td>appname/actions/</td>
<td>appname/devices/</td>
<td>588</td>
</tr>
<tr>
<td>appname/overview/</td>
<td>appname/utilization/</td>
<td>388</td>
</tr>
</tbody>
</table>

Step II

In order to answer more complex questions, we enriched the input data of the previous step by the knowledge that we gained in the previous step and used that as the input data for step II. We specifically focused on the navigation paths. For one, we constructed a network graph that visualizes the page views as well as their attributes (number of calls and maximum, minimum, average duration) as nodes and their relationships (navigation paths) as edges (see Figure 3-3). In order to being able to predict the user behavior, we used a Markov chain [Gagniuc, 2017] to statistically model the navigation flows and their probabilities (see Figure 3-4). As a result, the product owner gains knowledge on the behavioral patterns of his application’s users, such as the probabilities of choosing certain paths, including a prediction model for the next steps taken by a user. By implementing this step in real-time, the product manager can even observe dynamic changes in user behavior, for instance after a new deployment, in order to detect or react to occurring abnormalities.

Step III

Analogously to step II, we enriched the existing input data (of step II) by the generated knowledge of the previous step in order to answer the product owner’s most strategic question on optimization potentials. In this case, we followed the approach proposed by Singh and Kaur [Singh and Kaur, 2014]. They apply an ant colony based
Figure 3-3: Network graph of navigation paths

Figure 3-4: Navigation paths modeled as Markov chain
algorithm followed by a local search strategy in order to optimize website structures (see Figure 3-5). Ultimately, this results in concrete suggestions for the product owner how to adapt the website structures in order to improve the navigation flows of the application. By iteratively repeating the three steps implemented in this case, the product manager gets immediate feedback on how the optimized website structures, or any changes related to the measured metrics, influenced user behavior, and if any further actions might be recommended. The procedure, therefore, enables a continuous feedback loop from low level metrics to high level insights.

![Figure 3-5: Website structure optimization](image)

### 3.4.2 Case B

The procedure of Case B is described in detail in the following subsections. Figure 3-6 gives a first overview of the approach.

**Step I**

As the initial input data we received a set of application names as well as their class names and a list of platform APIs used by each class. Furthermore, each application class and each platform API belong to an application namespace or a platform namespace respectively. Each namespace can either contain several application classes or several platform APIs. We extracted the mappings between application classes, platform APIs, and their respective namespaces out of the data set and performed some basic aggregations in order to generate some first insights which APIs are used very frequently, on a regular basis, or only rarely (see Figure 3-7). This provides a first hint which APIs are important to ecosystem partners and, therefore, valuable for the platform.
Step II

On a more complex level, the product management was interested in common API usage patterns. We, therefore, enriched the available input data by the knowledge we gained in the previous step. We used the concrete mapping between application classes and platform APIs to generate a hierarchical clustering of the data set using Ward’s method [Ward Jr, 1963] in order to determine which APIs are commonly used in combination with certain other APIs (see table 3.2).

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>Platform API combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CompanyName.SubsystemA.ComponentB.API1</td>
</tr>
<tr>
<td></td>
<td>CompanyName.SubsystemA.ComponentB.API3</td>
</tr>
<tr>
<td></td>
<td>CompanyName.SubsystemC.ComponentA.API2</td>
</tr>
<tr>
<td>2</td>
<td>CompanyName.SubsystemA.ComponentA.API5</td>
</tr>
<tr>
<td></td>
<td>CompanyName.SubsystemB.ComponentA.API1</td>
</tr>
<tr>
<td></td>
<td>CompanyName.SubsystemB.ComponentB.API4</td>
</tr>
<tr>
<td></td>
<td>CompanyName.SubsystemB.ComponentB.API2</td>
</tr>
<tr>
<td>3</td>
<td>CompanyName.SubsystemB.ComponentA.API2</td>
</tr>
<tr>
<td></td>
<td>CompanyName.SubsystemC.ComponentA.API1</td>
</tr>
</tbody>
</table>

Table 3.2: Common platform API combinations per cluster

Step III

Lastly, we used the clustering results obtained in step II as the input data for step III. We trained a model based on the clustering which can be used to predict the cluster that an application class and its corresponding platform APIs belong to. By comparing the platform APIs of the respective cluster to the actual APIs used by the application class, the platform provider can improve its offerings by providing
tailored recommendations to the app providers which other platform APIs might be of interest to them. By continuously executing steps I to III in an automated pipeline, the recommendations get more precise with each iteration, and the platform provider can observe how the utilization of platform APIs evolves over time.

3.4.3 Case C

Lastly, the steps of Case C are described in the following subsections. Figure 3-8 provides an overview of the procedure.

Step I

As input data for Case C we received audit logs of the platform containing every operation executed over a period of two months. Each entry in the audit logs contains the operation ID, operation specific info (e.g., device IDs), a timestamp, and the user ID. Quite similar to the previous cases, we initially performed some basic data aggregations. We investigated the number of executions for each operation in order to determine which operations are quite common or frequent and, on the other hand, which operations are rather rare. By grouping the operations by timestamps and user IDs we were able to reconstruct the order of operation calls.
Step II

Building upon the knowledge of frequent single operations in step I, we aimed at analyzing the frequency of operation sequences in step II. We combined the operation IDs as well as the order of operations to recognize common sequence patterns. Figure 3-9 shows a snippet of the detected patterns. This can support the product managers’ understanding of how their system is actually used and what seems to constitute the most important tasks or functionalities for the users.

"Subset recognized 152 times: CREATE_USER_SUCCESS, ASSOCIATE_USER_ROLE_SUCCESS, ASSOCIATE_ROLE_WITH_USER"

"Subset recognized 75 times: ASSOCIATE_USER_ROLE_FAILURE, CREATE_USER_SUCCESS, ASSOCIATE_ROLE_WITH_USER"

"Subset recognized 152 times: ASSOCIATE_USER_ROLE_SUCCESS, ASSOCIATE_USER_ROLE_FAILURE, ASSOCIATE_USER_ROLE_SUCCESS"

Figure 3-9: Common operation sequences

Step III

The final goal was to utilize recurring patterns in order to generate recommendations to simplify the usage of the application. In a first step, we used the operations ID as well as the operation specific info to identify sequences of user role assignments to the same user. Next, we compared common combinations of user roles in order to generate recommendations on which roles can be consolidated into an overarching role.
comprising several sub-roles. Ultimately, this enables product managers to improve the offering of their product by reducing the number of operations for the users. By iteratively executing this procedure, product managers are able to monitor how the proposed changes affect the API usage rates and the behavioral usage patterns.

3.5 Results

Based on the described cases, we derived a generic model that describes a step-wise approach to achieving a quantitative measurement of high level insights for different levels of complexity. The derivation process is divided in two steps: First, we need to identify relevant and appropriate dimensions that constitute the overall frame for our model. Second, we extract and compare the steps taken in each procedure in order to deduce a generic process that we map to the dimensions defined in the previous step.

3.5.1 Dimensions

The first dimension of our model is straightforward since each of the steps taken in all of the cases started out with a set of input data that was further processed in order to generate some kind of valuable information. Therefore, the process of having data, utilizing it, and gaining actual insights constitutes the first dimension of our model. Moreover, the complexity and time horizon of questions, methods and insights increased with each of the steps performed during the procedures. The initial insights requested by the product managers were usually very simple, e.g. "Which features are most frequently used by my customers?", and can therefore often be answered by some simple aggregations of low level metrics. While on a second level the product managers really try to understand what the insights gained in the previous step actually mean, e.g. "How are users navigating to feature X or through the application in general?". Ultimately, the last phase in all of our cases was dedicated to improve or optimize a certain situation based on a previously observed behavior, e.g. "How to optimize the website structure in order to increase the efficiency of navigation flows?". Inspired by the three levels operational, tactical, and strategic applied in several fields such as planning [Hartzell, 2018], decision making [Chand, 2018], and even dashboards for system monitoring [Durcevic, 2018], we define the different phases – that constitute the second dimension of our model – as follows:

1. Operational: aims at answering immediate questions ("what?") and providing basic insights by using solely low level metrics to describe a current state or situation (e.g. comparison of current feature usage rates); The goal is to provide fast feedback of current behavior

2. Tactical: aims at interpreting data ("how?") and describing certain phenomena, behavioral patterns, or unexpected behavior by combining low level metrics with the previously gained knowledge and applying advanced techniques (e.g. clustering of APIs that are often used in combination); The goal is to identify
3. Strategic: aims at utilizing the data in order to provide further suggestions, explanations or interpretations that are beyond just a tactical representation of data ("how to...?"); the goal is to increase the business or customer value in the future by providing continuous suggestions for optimizations or concrete action items to increase business/customer value (e.g. website structure optimization based on navigation flows).

Figure 3-10a shows the dimensions as well as the process steps of our model.

3.5.2 Process

The entry point for all of our cases was an initial set of data containing low level metrics. After getting a first overview, we entered a planning phase in which we identified the relevant metrics, performed some data cleaning operations, and lastly defined the concrete measurements that are necessary to address the requested insights or questions. The latter were identified in collaboration with representatives of the respective cases. Following this, the next step contains the execution of the previously defined measurements and the visualization and processing of the results into knowledge. The latter is also the output of the first iteration and commonly contains the behavior of single metrics and the interpretation of their behavior, e.g. if the usage rates of a feature are high, it is considered important.

In the second iteration, the input data is enriched by the knowledge gained in the previous iteration. In the planning phase, the features or feature combinations within the input data that describe a certain phenomena are identified and appropriate methods are selected in order to generate representations of data that address the requested insights. Next, these methods are applied to the data set and the results are processed into knowledge. It can, for instance, contain the representations of certain situations.
or behaviors in the data set (e.g. as clusters) or the interpretation of these representations (e.g. understanding how a user navigates through an application based on a probability model).

This newly derived knowledge is, analogously to the previous iteration, added to the input data of the third iteration. The following planning phase comprises the definition of (optimization) goals based on the tactical knowledge as well as the selection of the appropriate algorithms or techniques to do so. These techniques are then applied to the data set in order to generate some kind of recommendations or explanations. Ultimately, the final output can, for instance, be an explanation for a certain behavior or even suggestions or recommendations for possible optimizations. In summary, the main steps in each iteration are: data input, planning, execution & interpretation, and knowledge (see Figure 3-10b).

3.5.3 Model

As a final result, we mapped the steps of the process to the dimensions described in the beginning of the section in order to introduce a generic model that enables a quantitative measurement of product management insights (see Figure 3-11). The model consists of three levels; operational, tactical and strategic. There is a standardized process defined that is passed through in each level. The input data resembles the basis for each level, followed by a planning phase in which the necessary data as well as the required operations (e.g. measurements, algorithms or techniques) are specified. In the next step, the respective operations are executed and the results are interpreted and processed into knowledge. For one, the knowledge itself already provides product management insights on different levels (operational, tactical, strategic), for another the knowledge gained in previous levels is used as input data in the higher levels.

The key principle of our model is the strategic concatenation of the individual levels in order to reach actual business level insights. The operational level mostly includes the description of predefined KPIs or basic interests. Its main advantage is the capability to provide fast and immediate feedback of the system's current behavior. Even unexpected behavior will most likely first show up on this level since it is closest to the actual data. The tactical level builds upon the operational level and aims at highlighting and monitoring the current value of, for instance, a feature or behavioral patterns. By combining multiple data sources, the tactical level provides a deeper understanding of how certain behaviors occur and which relationships might exist. This enables product managers to comprehend actual user behavior and to base and monitor their decisions on these insights. Building upon this, the strategic level aims at analyzing the current value situation in order to identify further suggestions, explanations or interpretations that can, in the future, increase the business or customer value. The product manager continuously receives suggestions and concrete action items for anything related to business or customer value, such as optimizations, simplifications or recommendations. The overall vision is to automate this chain-like process in order to enable continuous, real-time measurement of high level product management insights based on quantitative data.
3.6 Related Work

This section provides an overview of existing research in the field of measuring high level insights, such as customer or feature value, based on quantitative data. Emeakaroha et al. [Emeakaroha et al., 2010] developed a framework to monitor high level service level agreements (SLAs) based on low level metrics. Their framework assumes that SLAs are agreed upon and threat thresholds are set. Next, resource metrics are measured and mapped to SLA instances based on predefined mapping rules. Mapping rules can either be simple (e.g. one-to-one mapping of metric to SLA) or complex (e.g. several resource metrics are used to calculate SLA parameters based on a formula). A runtime monitor periodically receives the low level metrics, applies the mapping rules and compares it to the SLAs and their defined threat thresholds [Emeakaroha et al., 2010]. In contrast to our approach, this work focuses more on the individual components and infrastructure that is needed to achieve the mapping, instead of the actual mapping process. The mapping rules described in their work solely include numeric values and straightforward formulas to represent the SLAs, whereas our model is designed to enable mappings to different levels of abstractions by applying more advanced data analytic techniques and including multiple types of data sources (e.g. textual information).

According to Butz and Goodstein [Butz Jr. and Goodstein, 1996] it is important gain a profound understanding of the customers that use a product, going beyond customer satisfaction, in order to fully comprehend customer value. They provide a checklist for a customer understanding process that, among others, includes: a) How does this customer use our product? b) How could our product be easier for this
customer to use? c) How could we expand our service(s) to reduce this customer’s
problems? [Butz Jr. and Goodstein, 1996] This catalogue of questions served as an
orientation for the definition of our cases and the formulation of the high level insights
respectively.

In a previous study, Marciuska et al. [Marciuska et al., 2013] observed that a high
feature usage relates to a high perceived customer value, while a low usage does not
necessarily relate to a low customer value. Based on these insights they developed
an approach that combines different usage metrics in order to determine a feature’s
value. First, they monitor the relative actual feature usage (e.g. the number of times
it is used) before determining a threshold between valuable and non-valuable features
based on the perceived values of the participants in their study. Since these metrics
were not sufficient to determine valuable features that are rarely used, they addition-
ally asked the developers about their intended feature usage estimations. Ultimately,
they combined the relative actual usage, the intended usage, and the usage threshold
to provide an indicator for non-valuable features in order to support decision making
processes [Marciuska et al., 2014]. Especially for the operational level in our model it
is important to consider that low usage rates do not necessarily imply a low customer
value.

There exist three different types of analytics that organizations can apply to gain
insights on their business. Descriptive analyses provide the summary of a certain
condition by looking at data sets from the past. Predictive procedures are used to
predict future behaviors or events, while prescriptive techniques investigate future
scenarios, trying to find "a solution or insightful actions from the predictions" [Saggi
and Jain, 2018]. Depending on the variables used during the operations, measure-
ments can either be direct (one variable) or indirect (multiple variables) [Kaner et al.,
2004]. The operational level in our model is mainly designed for descriptive analyses
using direct measurements, while the tactical level includes predictive techniques us-
ing indirect measurements, and the strategic level utilizes prescriptive methods with
indirect measurements.

A very common method to measure the impact of certain changes is A/B experi-
mentation [Kohavi and Thomke, 2017], [Fabijan et al., 2017a]. Two versions, A (control)
and B (treatment), of the same product are deployed and the users get randomly
assigned to one of the versions. Previously defined key metrics ("Overall Evaluation
Criteria"), e.g. sales or click rates, can then be compared [Kohavi and Thomke, 2017].
This allows the identification of the value of a product or feature, and the changes
that contribute to it [Fabijan et al., 2017a]. However, defining the overall evaluation
criteria as well as aligning business KPIs and team level metrics have been identified
as key challenges with A/B experimentation [Olsson et al., 2017], [Fabijan et al.,
2017b]. This challenge could be addressed by integrating the approach proposed in
our model into the experimentation process.
3.7 Conclusion

With increasing efforts to reduce the time-to-market of software products additional challenges arise. In order to improve their products, product managers often want to understand how their software is used by their customers. However, since these customers are continuously exposed to new versions of a product, their perception of the product as well as the customer value of certain features changes dynamically. Therefore, the results of qualitative studies can be outdated quite fast. As a result, the need for alternative ways to continuously measure high level business insights arises.

In this paper we propose a model that enables automated, real-time measurement of high level business insights based on quantitative data. We derived the model from three real-world cases that we implemented in three different platforms or products. The main concept of our model is to iteratively increase the complexity of generated insights by using the knowledge gained during one iteration as input data in the following iterations. We structured the different types of requested insights into three levels, operational, tactical and strategic, that each build upon the results of the previous level and serve as one dimension in our model. The second dimension represents the process of turning data into valuable knowledge, starting out with a set of input data, followed by a planning phase, the execution of the selected methods and interpretation of the results, and resulting in a set of generated knowledge that is used as input data in the next iterations. This process is repeated for all of the levels.

Each of the vertical layers in our model can be implemented as automated execution threads and by using the output of one layer as the input for the next layer, an interface between two layers emerges that enables the concatenation of all layers. By building these layers upon each other, our model enables a step-by-step increase of complexity for the generated high level insights. Ultimately, these high level business insights can be quantitatively measured in real-time by running the process described in our model as an automated pipeline.

The number of layers that are executed in the pipeline is flexible and can be adapted to a different number of requested insights and different degrees of complexity. Our model mainly addresses product management insights but future work could also be dedicated into extending the model to different roles of the platforms and products, or organizations.
Chapter 4

Customer Churn Prediction in B2B Contexts


Abstract

While business-to-customer (B2C) companies, in the telecom sector for instance, have been making use of customer churn prediction for many years, churn prediction in the business-to-business (B2B) domain receives much less attention in existing literature. Nevertheless, B2B-specific characteristics, such as a lower number of customers with much higher transactional values, indicate the importance of identifying potentially churning customers. To achieve this, we implemented a prediction model for customer churn within a B2B software product and derived a model based on the results. For one, we present an approach that enables the mapping of customer- and end-user-data based on "customer phases" which allows the prediction model to take all critical influencing factors into consideration. In addition to that, we introduce a B2B customer churn prediction process based on the proposed data mapping.

4.1 Introduction

Data on customer behavior can provide valuable insights on future decisions made by a customer. Churn prediction models, for instance, identify customers "who stop using a product or service" [Sajjadi, 2018]. This is of high interest to product providers since a large number of churning customers not only leads to a loss of revenue but can also have a negative impact on a company’s reputation [Verbeke et al., 2011]. While the field of customer churn prediction is well-researched in the business-to-customer (B2C) domain, it receives much less attention in business-to-business (B2B) contexts [Jahromi et al., 2014]. The number of customers in B2B businesses is usu-
ally significantly lower but their transactional values are often a lot higher. Therefore, single customers are of high value to a company and the impact of losing one is much bigger [Stevens, 2005]. This backs up the relevance of customer churn prediction in B2B contexts. However, approaches developed for B2C systems can often not be applied in B2B environments due to their complex setups. For instance, the customer buying a product is not necessarily the actual user of the product. While the customer makes the final decision of a purchase, the decision is to some extent influenced by the end-users.

For this reason, we conducted an exploratory study to answer the following research questions: 1) How can customer churn be predicted in B2B contexts while taking B2B-specific characteristics into consideration?; and 2) How can customer data as well as end-user data be combined in order to take all influencing factors into consideration?

In order to address these questions, we implemented a customer churn prediction model in a real-world product and derived the approach presented in this paper from the instantiation of the respective solutions. Specifically, we developed an approach that enables the mapping of end-user and usage data to customer data based on so called customer phases resulting in a shared data set that forms the basis of the prediction process. The shared data set is then used as the input for the prediction model itself.

The contribution of this paper is two-fold: First, we provide an approach for overcoming the challenge of combining and mapping data of different stakeholders who, either directly or indirectly, influence a certain decision, such as the purchasing behavior of a product's customers. Second, we present a step-wise process that enables the prediction of customer churn in a B2B context based on customer- as well as end-user-data by using the previously mapped data as input for the prediction model.

The remainder of this paper is structured as follows: Section 4.2 gives an overview of related work in this area, before we elaborate the research method as well as the research context in Section 4.3. In Section 4.4, we describe the implementational details of our approach. The results of our study are presented in Section 4.5, followed by a conclusion in Section 4.6 including a discussion on the generalizability of our approach.

4.2 Related Work

Oftentimes, subscriber data required for churn prediction models changes dynamically over time. This results in the need to retrain prediction models on a regular basis in order to "overcome data staleness and inconsistency" [Yan et al., 2004]. Moreover, most data sets in this area of applications are highly imbalanced in relation to class distribution. Precisely, the rate of accepting an offer is often much lower than the rate of a declined offer [Yan et al., 2004]. Since we discovered the same characteristics in our data set, we resampled it to overcome the imbalance as proposed by [Yan et al., 2004] or [Wang et al., 2016]. Additionally to retrain the model iteratively, we propose an approach based on different customer phases that depict the commonly changing
behavior for each customer over time in the purchasing process. Ullah et al. [Ullah et al., 2019] propose an approach for customer churn prediction in the telecom sector (B2C) that additionally provides the reason or factors behind the churning of customers in order to derive retention strategies. Lastly, the k-means algorithm groups the customers into one of three categories (low, medium, or high) resembling their respective risk of churn. Recommendation systems can then propose strategies that worked on similarly behaving customers, using a collaborative approach [Ullah et al., 2019]. In order to being able to act upon certain predictions, it is critical to know and understand the underlying factors of a churning customer. However, the perceived value of a product differs between B2C and B2B customers [Mencarelli and Riviere, 2015]. We, therefore, decided to tag the identified factors as either customer factors or end-user factors.

While the majority of studies in this area investigate business-to-customer (B2C) relationships, Kandeil et al. [Kandeil et al., 2014] use the outcome of the LRFM (Length, Recency, Frequency, Monetary) analysis [Hughes, 2005], [Chang and Tsay, 2004] to cluster customers in a business-to-business (B2B) setting into different categories. These results can be used as a basis for customized marketing strategies [Kandeil et al., 2014]. Related to this, Jahromi et al. [Jahromi et al., 2014] use data mining techniques to predict churn of customers in a B2B non-contractual environment. Based on their predictions, they developed a retention campaign to maximize a company’s profit. The case product in our study is in the contractual domain and the type of features used for classification go beyond the core customer behavior resembled by LRFM but also include the end-users’ interaction with the product.

4.3 Research Method & Case Context

We chose to inductively derive the approach presented in this paper from instantiating it in a real-world product [Goddard and Melville, 2004]. Specifically, we interviewed three stakeholders of a B2B software product provider in order to identify their challenges related to B2B customer churn prediction. Additionally, we examined their database and data structures to get an understanding of B2B-specific data characteristics. We strengthen the generalizability of these characteristics by comparing them to other B2B cases that we have studied in earlier publications (in [Figalist et al., 2019c] and [Figalist et al., 2019a]). In a next step, we developed and implemented an approach to predict customer churn in B2B contexts while addressing the previously identified challenges and characteristics. We extract the characteristics that our approach is based upon as well as the steps we have taken during the instantiation to build a generic model for B2B customer churn prediction taking B2B-specific factors into consideration. We validate the generalizability of our model by showing that each of the characteristics and single components has also been observed in other B2B products we have worked with or have been applied in a similar context in the literature.

The product itself is the platform of a software ecosystem that is established in the healthcare domain. Multiple platform-internal as well as external applications are
developed based on the platform. The platform provider offers a variety of licensing options to its customers, including basic and premium licenses as well as trial phases. The decision about a purchase is made by the respective customer, while the users of the product are typically the customer’s employees. The decision is, therefore, directly influenced by the customer and indirectly influenced by the end-users of the product.

4.4 Implementation

In order to derive an approach for customer churn prediction in B2B contexts, we implemented a churn prediction model for a real-world B2B software product (see Section 4.3). We start by identifying questions and hypotheses related to churn, before exploring the available data and generating a shared feature set comprising both customer- and end-user data. Finally, we preprocess the data set, train the prediction model and evaluate and interpret the results. All (case-specific) implementation details of our approach will be provided in the following subsections.

4.4.1 Data Preparation

Before starting to implement the prediction approach, we conducted several unstructured interview sessions with a product owner, an operations engineer, and a data analyst of the platform’s development teams. They provided us valuable insights on the available data, the platform provider’s interactions and relationships with their customers as well as important events in the lifecycle of each customer. During the interviews the platform provider revealed a strong interest in the evolution of customer and user behavior over time as well as the impact of features in different points in time. Therefore, we start by identifying the steps or phases that each customer goes through from registration to making a decision (churn vs. non-churn). Next, we process and extract the available platform data, before mapping it to the defined phases.

Customer Phases

In order to identify all relevant events that constitute the frame for the phases, we interviewed three stakeholders of the platform who provided us insights on the important events and phases each customer goes through from registration to the decision on whether to stick with a premium license or not. As a result, the first phase "Onboarding" covers the interval between the registration date of a customer and the effective date of that customer’s basic license. At any later point in time, the customer can choose to enter a trial period during which its users can experience the premium features of a product. The second phase "Basic", therefore, comprises the time between a customer’s basic license effective date to the starting date of the trial period of that specific customer. At the end of that trial period, the customer needs to decide whether to keep the premium version of the product (non-churn) or
whether to go back to the basic license (churn). Additionally, one of the interviewees' hypothesis was that the last couple of days or week before the trial period expires have a greater impact on the decision than the weeks before that. For this reason, we decided to split up that timeframe into two phases: "Trial" and "Pre-Decision". The pre-decision phase starts ten days before the trial period expires. In order to validate the defined phases, we presented them to the interviewees once again who approved the described customer phases.

**Data Extraction**

The product's database stores four different types of data. For one, each table is either related to the customer or the end-user of the product. Second, the data can either be static (e.g. a customer's registration date) or dynamic (e.g. number of logins per day). Moreover, multiple measures (direct metrics) can be combined to generate derived metrics that also hold valuable information. Table 4.1 gives an overview of all extracted features organized by its type.

In order to generate a mapped data set that constitutes the foundation of our prediction approach, we define the time attribute as the main mapping criterion, specifically the customer phases. In a first step, all static features identified during the data selection and related to either the customer or the user of a customer are extracted and linked to the respective customer ID. Following this, the timeframes of each of the defined customer phases are extracted for each customer individually to generate a customer-specific timeline. Based on this, each of the dynamic features are extracted for each of the computed customer phases, and are, again, linked to their respective customer ID.

<table>
<thead>
<tr>
<th>Customer Data</th>
<th>User / Usage data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static</strong></td>
<td><strong>Dynamic</strong></td>
</tr>
<tr>
<td>Direct</td>
<td>purchase info</td>
</tr>
<tr>
<td>Derived</td>
<td>avg. downtime of receiver</td>
</tr>
</tbody>
</table>

| Extraction Type | Once | Per phase | Once | Per phase |

Table 4.1: Extracted features per type

**4.4.2 Prediction Model**

In order to train our prediction model, we preprocessed the mapped data set into a labeled input data set by removing rows with missing values, standardizing the columns and adding a label for each customer on whether they downgraded to the basic license or not (binary classification).

We apply a feature selection technique [Dash and Liu, 1997] to the data set in order to identify the most relevant features for training the prediction model. As a result, 25 out of 50 features are selected for further processing. The input data set is split
up into a training data set (80%) and a test data set (20%). Next, we build a neural
network using the 25 selected input features, one target variable and two hidden layers
with twelve nodes each. We set the number of epochs to 30 and the batch size to ten
before training the model. Finally, we evaluate our model by predicting the output
variable of the test data set, achieving an accuracy of 88% with a precision of 0.89
and recall of 0.91.

4.5 Derived Data Mapping Model and Prediction
Process

On an abstract level, the implementation of our study relies on two core mechanisms:
1) the generation of a mapped data set that combines customer as well as user/usage
data by linking their respective behavior to shared customer phases; and 2) the con-
struction of a process for predicting customer churn decisions based on the previously
generated data set. Based on our implementation, we derive a generic model for
both of these mechanisms which will be explained in a greater detail in the following
subsections.

4.5.1 Data Mapping

One important characteristic of B2B businesses is the differentiation between cus-
tomer and end-user. While the customer is behind the purchase of the product, the
customer’s customer or employee is the one using it [Kandeil et al., 2014]. The benefits
and, therefore, perceived value of a product or feature is different for each of the
stakeholders [Mencarelli and Riviere, 2015]. In previous studies [Figalist et al., 2019c],
[Figalist et al., 2019a] we worked with four different platform providers established in
various B2B businesses, including the one studied in this paper. While investigating
the communication structures, we found that even though the customers and end-
users are two individual entities, they still share direct communication channels. As
a result, an end-user’s dissatisfaction might indirectly influence a customer’s decision
to renew, or not renew, a contract. We, therefore, argue that it is beneficial to also
take end-user data into consideration when implementing churn prediction models
for B2B businesses. This, however, results in the need for a shared data structure or
data mapping. We propose a model that enables the mapping of customer- as well
as end-user-data based on customer phases. For any type of product, customers usu-
ally trigger a series of product-specific events (e.g. registration, purchase, renewing
a contract etc.). Customer phases are the timeframes between such events and serve
as a basis for our mapping approach. The approach itself consists of three steps: the
definition of customer phases, data selection, and data processing.

Step I: Definition of Customer Phases

Using customer phases as the main mapping criterion has two advantages. For one,
it makes it easy to link customer as well as end-user behavior to each phase simply
based on the timestamps. In addition to that, it can portray changes in behaviors over time and include these changes in the prediction model. In order to define the customer phases, we start by identifying the steps or phases that all customers go through before they make a decision to churn or not to churn. Oftentimes a series of events serves as the frame for the phases.

**Step II: Data Selection**

On an abstract level, each feature in the data set can be characterized by three different attributes. For one, a data point can either be related to the customer or the end-user. Furthermore, each data point can either be characterized as static or dynamic. Static data holds information that does not change over time. Dynamic data points, however, are dependent on their either pre-defined or event-driven recorded point of time. Lastly, some data points can be extracted as direct metrics, while other data points can be combined and further processed into derived metrics.

**Step III: Data Processing**

Based on the preceding steps, the data is now processed into a shared data model (see Figure 4-1). We define the time attribute as the main mapping criterion, specifically the customer phases. First, all static features related to either the customer or the user of a customer are extracted and linked to the respective customer ID (see yellow boxes at top & bottom in Figure 4-1). Derived metrics are computed by further processing one or more direct metrics. Following this, the timeframes of each of the defined customer phases are extracted for each customer individually to generate a customer-specific timeline (see Customer Phases in Figure 4-1). Based on this, each of the dynamic features are extracted for each of the computed customer phases (see green boxes above and below customer phases in Figure 4-1). After a customer passes through all the phases, it needs to decide for or against churning.

![Figure 4-1: Overview of the mapping approach based on customer phases](image)

**4.5.2 Prediction Process**

The previously generated mapped data set is crucial to the entire prediction process since it serves as the input data of the procedure. Initially, the input data set is
turned into a labeled data set by a) mapping previous decision outcomes (churn or non-churn) to the respective customer ID; b) cleaning, standardizing, and resampling the data; and c) applying appropriate feature selection techniques [Dash and Liu, 1997] to identify the relevant feature set. Based on the labeled data set, a classification technique can be applied in order to train a prediction model.

In order to enable customer churn prediction in B2B contexts, we combine our data mapping approach with the described prediction procedure. Figure 4-2 shows the generic model we derived during our study. It consists of the following steps (linked to numbers in model): S1-3: Relevant metrics are identified and extracted by following the data mapping approach presented in Section 4.5.1. All metrics based on static data are extracted once for each customer while all metrics based on dynamic data are extracted for each phase and customer. All extracted metrics are combined to a shared data set. S4-5: The previously generated data set is then labeled (churn vs. non-churn) based on the previous decision of the customers. After preprocessing (standardizing, resampling, feature selection etc.) the data set, it can be used as the input to train a prediction model. Ultimately, our model enables practitioners to predict customer churn based on customer- as well as end-user-data, thereby taking all influencing factors into consideration.

4.6 Conclusion

Single customers of B2B businesses are often of greater importance compared to B2C businesses since their number is typically much lower [Stevens, 2005] but their transactional value is much higher [Rauyruen and Miller, 2007]. Losing even one might have a significant impact on the provider of B2B products [Stevens, 2005]. While this reinforces the importance of customer churn prediction in B2B contexts, there is a lack of research on how to achieve this [Jahromi et al., 2014]. The goal of this exploratory study was to investigate how to perform customer churn prediction in B2B contexts while taking B2B-specific characteristics into consideration. We implemented this in a real-world product and derived a two-stage process.
that consists of building a shared data model as well as the prediction process itself. During the data mapping, a shared data set of customer- as well as end-user-data is generated based on customer phases. This data set is then used as input for the prediction model.

One of the limitations and threat to validity of this study is the number of investigated cases. However, after working with multiple other B2B product providers prior to this study (e.g. in [Figalist et al., 2019c] and [Figalist et al., 2019a]), and comparing the B2B-specific characteristics to the ones identified in existing literature (e.g. in [Kandeil et al., 2014], [Stevens, 2005], or [Rauyruen and Miller, 2007]), we have evidence to believe in the generalizability of the presented approach. Moreover, we plan on implementing this approach for other B2B products in order to further validate our model.
Chapter 5

Fast and Curious: A Model for Building Efficient Monitoring- and Decision-Making Frameworks Based on Quantitative Data


Abstract

Context: Nowadays, the hype around artificial intelligence is at its absolute peak. Large amounts of data are collected every second of the day and a variety of tools exists to enable easy analysis of data. In practice, however, making meaningful use of it is way more challenging. For instance, affected stakeholders often struggle to specify their information needs and to interpret the results of such analyses.

Objective: In this study we investigate how to enable continuous monitoring of information needs, and the generation of knowledge and insights for various stakeholders involved in the lifecycle of software-intensive products. The overarching goal is to support their decision making by providing relevant insights related to their area of responsibility.

Method: We implement multiple monitoring- and decision-making frameworks for six individual, real-world cases selected from three different platforms and covering four types of stakeholders. We compare the individual procedures to derive a generic process for instantiating such frameworks as well as a model to scale it up for multiple stakeholders.

Results: For one, we discovered that information needs of stakeholders are often related to a limited subset of data sources and should be specified in stages. For another, stakeholders often benefit from sharing and reusing existing components among themselves in later phases. Specifically, we identify three types of reuse: 1) Data and
knowledge, 2) tools and methods, and 3) concepts. As a result, key aspects of our model are iterative feedback and specification cycles as well as the reuse of appropriate components to speed up the instantiation process and maximize the efficiency of the model.

Conclusion: Our results indicate that knowledge and insights can be generated much faster and stakeholders feel the benefits of the analysis very early on by iteratively specifying information needs and by systematically sharing and reusing knowledge, tools, and concepts.

5.1 Introduction

In recent years, the prominence of data-driven approaches in software engineering increased tremendously. Data emerges from many different sources throughout the entire software engineering process and a variety of analysis techniques is available to basically anyone.

Software analytics enables practitioners to derive high-value information from low level metrics [Zhang et al., 2013]. It can be used to either verify or disprove hypotheses made by humans or to continuously monitor and check software systems [Menzies and Zimmermann, 2018]. With increasing amounts of data, manual analyses become more and more complex and time-consuming, whereas software analytics can be used to automate such processes [Krishna et al., 2018].

However, companies are still struggling to make actual and meaningful use of data [Bizer et al., 2012], [Olsson and Bosch, 2013]. For one, it is challenging to keep track of collected data and the purpose it is collected for [Olsson and Bosch, 2013]. For another, the "difficulty of interpreting data is a significant barrier to the use of analytics today" [Buse and Zimmermann, 2012]. Additionally, companies often experience difficulties in establishing a systematic sharing process within their organizations [Fabijan et al., 2016]. As a result, operational data is oftentimes not shared properly within an organization to support decision making processes [Cito, 2016].

In previous research, a gap between the information needs of stakeholders and available data has been identified [Buse and Zimmermann, 2012], [Figalist et al., 2019a]. It is crucial to provide a comprehensible explanation for the connection between low level data and high level information needs. This supports stakeholders in understanding certain phenomena and making informed decisions [Buse and Zimmermann, 2012], [Dam et al., 2018].

Moreover, the "overarching goal of analytics is to [...] move beyond information and toward insights" [Buse and Zimmermann, 2012]. However, a simple one-to-one mapping between high level insights and the collected quantitative data is usually not feasible. Rather, a whole range of methods needs to be applied in order to reach a certain degree of complexity, accuracy and trustworthiness to describe the required high level information needs. This fact significantly impedes the implementation and automation of such a process.

The study presented in this paper aims at addressing these challenges by investigating how quantitative data can be utilized to systematically support stakeholders in
their everyday work. To achieve this, we select six cases out of three real-world product teams covering four types of stakeholders (product management, operation engineers, software architects, and sales). We implement an approach for each of the cases to monitor stakeholder-specific information needs and generate knowledge to support their decision making. By comparing the individual procedures, we are able to derive a generic model that enables efficient realizations of such monitoring- and decision-making frameworks for multiple stakeholders.

The contribution of this paper is three-fold. First, we explore the information needs of different stakeholders involved in the lifecycle of a software product and present the flow of data and knowledge required to address these needs. Furthermore, we provide a process for instantiating monitoring- and decision-making frameworks for individual stakeholders based on the utilization and analysis of quantitative operations data collected during the runtime of a system. Specifically, the process enables practitioners to extract useful information, detect insightful patterns, and generate strategic knowledge. The overarching goal is to provide the necessary understanding of available data as well as underlying explanations of its behavior in stakeholder-specific areas of responsibility to, ultimately, support their decision making. In addition to that, we derive a generic model that enables an efficient realization of this process for multiple stakeholders. By focusing on primary data sources at the beginning and fostering the reuse of components, tools, and concepts later on during the analysis, our model can increase the efficiency of the analysis and, thereby, create value to stakeholders very fast and early on.

The remainder of this paper is structured as follows: In Section 5.2 we outline the background of our study. Next, the study design is presented in Section 5.3, followed by an overview of the related work in Section 5.4. Section 5.5 describes the implementational details of our study while the results are presented in Section 5.6. Finally, we conclude our work in Section 5.7.

5.2 Background

The term software analytics (SA) describes analytics performed on software data to generate valuable insights for various stakeholders, from managers to software engineers, to ultimately support their decision making [Buse and Zimmermann, 2012], [Menzies and Zimmermann, 2013]. There are four types of analytics that can be performed on the data: 1) Descriptive analytics (simple summaries of and information on past behavior or performance); 2) diagnostic analytics (explanations of and insights about the past); 3) predictive analytics (predictions of future outcome based on current or historical data); and 4) prescriptive analytics (recommendations of actions) [Nayebi et al., 2015].

In their systematic mapping study, Nayebi et al. [Nayebi et al., 2015] identified a gap in existing literature between the type of analytics addressed in research papers (predictive) and the importance of types in practice (descriptive). Additionally, the authors highlight that while it is important to first understand the past events before providing any predictions or prescriptions, it is not yet reflected in the literature.
Moreover, the main stakeholder being addressed by SA research is software development, while other stakeholders such as project or product management, sales, or marketing are being neglected in existing literature [Abdellatif et al., 2015], [Hassan et al., 2013]. Closing this gap by also covering tasks of other stakeholders could enable SA "to become a powerful, strategic, decision-making instrument" [Hassan et al., 2013].

Another key issue is the applicability of results in practice. Software analytics needs to be actionable in order to constitute value for practitioners [Yang et al., 2017]. Especially the SA result of less researched stakeholders (e.g. product management) "is still immature and is similar to the direct statistics and dashboard work [Abdellatif et al., 2015]."

As a result to the identified gaps, our study aims at 1) covering multiple types of analytics (from descriptive to diagnostic to predictive and prescriptive); 2) investigating multiple, underresearched types of stakeholders (e.g. product management and sales); and 3) a close collaboration with the stakeholders to provide in-depth analyses and ensure relevance and applicability of results.

5.3 Study Design

We chose to inductively derive [Goddard and Melville, 2004] the approach presented in this paper from instantiating monitoring- and decision-making frameworks in multiple real-world cases. As our study aims at covering different types of stakeholders, we sent out a description and invitation to participate in the study to multiple contact persons of various platforms and products who shared it across their teams. We received responses from three different, real-world platform providers established in different domains and were able to identify six independent cases, covering ten stakeholders of four different types (product management, operation engineers, software architects, and sales). Each of the stakeholders works for one out of the three platforms or platform products and was selected due to his or her interest in working with data-driven monitoring- and decision-making frameworks.

5.3.1 Case Platforms and Study Participants

An overview of the stakeholders and their project contexts is presented in Table 5.1. In addition to that, a more detailed description of the platforms, the stakeholders and their use cases, as well as the available data is provided in the following subsections.

Healthcare Platform

*Cases A, D, and F* are based on a platform provider established in healthcare domain. The platform hosts several platform-internal as well as external medical applications. During the course of this study we worked with a product manager of one of the platform’s internal applications, two operations engineers, and a product manager.
representing the sales team. This intermediary function was necessary due to company restrictions.

The application’s product manager has more than ten years of experience in software engineering and product management, and has worked in this specific role for the past four years. The application visualizes information about the users hardware devices and offers several configuration and filter functionalities for these visualizations. The product manager would like to learn more about how the application is used and how it can potentially be optimized for its users.

The operations engineers have been working in their roles for three and one and a half years, and have an overall experience in software and operations engineering of five and two years respectively. They are responsible for operating the platform as well as all platform-internal applications. Whenever a service is not available to the users, it is reported as an incident. As a result, the operations engineers would like to know more about the occurrence of such incidents in order to being able to react timely or even avert the incident.

Lastly, the product manager representing sales has an overall experience of seven years in software engineering and product management, and spent the last three years working in his current role. As part of this role, he is responsible for building reporting dashboards for sales, is in regular contact with the sales team and knows their requirements very well. Therefore, we believe he is a suitable candidate to represent the sales team in our study. In order to reach out and contact customers more strategically, it is of particular interest to him and the sales staff how customers of different license types are behaving and whether there is a kind of behavior that is typical for churning customers.

The data provided by the platform provider includes: 1) incident data (service unavailability); 2) usage data (e.g. logins, page views, session information); 3) master data records (e.g. registration dates of customers and users); and 4) license data (e.g. license type, effective date, expiration date).

### Industrial Platform

For Cases B and E of our study we worked with a platform provider operating in the industrial domain. Several applications run on the platform and are developed by various divisions within the company. The platform and its applications offer functionalities for device configurations, implementation of device logic and commissioning of industrial devices including their networking. We collaborated with two software architects and two product managers of the platform.
The software architects have about eight and ten years of experience in software engineering and software architecture, and two and five years in their role as software architects of the platform. They are currently facing the challenge that the system grew tremendously over the past decade and as a result the API design became very complex and needs to be refactored. Therefore, they are interested in how the APIs are used by external developers in order to increase both, usability and maintainability, during refactoring.

The product managers have been working in their current role for two and one and a half years with a total of five and ten years of experience in product management. They are particularly interested in how external developers are currently using the APIs provided by the platform and how they can simplify the use of their APIs. We were given access to the platform’s 1) dependency data (dependencies between APIs); 2) static API data (reference of APIs used by external applications); and 3) master data records (information on external developers and their applications).

Remote Connection Provider

For Case C we collaborated with a platform that provides services to establish remote connections to industrial devices and plants. The product managers we worked with have eleven and twelve years of experience in software architecture and product management and have been working for the platform provider for three and two years respectively. The platform itself offers a variety of operations to different user roles (e.g. for user or configuration management) and, in addition to that, custom operations can be defined. Since this overload of operations can become very inconvenient for the users, the product managers would like to know more about the most frequent operations and operation sequences in order to decrease the system’s complexity. To investigate this, the platform provider gave us access to the audit logs storing detailed information about the operations executed by the users.

5.3.2 Research Process

The overall research process is outlined in Figure 5-1. We started by defining the overarching research question of our study: How to efficiently generate stakeholder-targeted information and insights based on quantitative data? Next, we reviewed existing literature in the fields of software analytics and data-driven measurement approaches to identify relevant research related to our research goal and the advances made in this area. To achieve this, we queried common scientific libraries (IEEE Xplore, ACM Digital Library, ScienceDirect, Springer Link) using search terms related to the respective fields: "software AND (analytics OR analytics data OR analytics stakeholder OR measurement OR measurement framework OR measurement metrics OR experimentation OR information needs OR data-driven development OR data-driven decision making)". Following this, we conducted some first initial interviews with each of the stakeholders participating in our study which lasted between 30 and 60 minutes. During these interviews, the available data sources were examined and discussed with the respective stakeholders. This helped the participants in
understanding what type of information is available in the data. Next, they were asked to explain and specify a first set of information needs that would be of interest to them. We discussed and refined the information needs in collaboration with the stakeholders, before documenting them for the implementation phase at the end of each interview.

The predefined information needs are used as the first input for our implementations. Based on the individual needs, we selected a subset of data sources related to the respective information need as well as one or more analysis techniques to generate meaningful results that answer or address the needs. This was followed by the actual implementation of analyses. Depending on the results, we adapted the input data sources or analysis techniques until the stakeholders’ requested needs and insights were satisfactorily addressed.

Once an implementation was completed, we presented the results to the respective stakeholders who were then asked to provide feedback. These feedback interviews lasted between 30 and 60 minutes. Specifically, we asked the participants 1) how well the results address their information needs; 2) if there is anything they would like to change or improve; and 3) what additional, more complex information needs or insights they would wish for based on the results. We discussed and documented the new information needs analogous to the initial interviews.

As part of an iterative cycle, another implementation phase was then triggered to revise existing implementations and to run additional analyses generating more high-level and complex insights. This was again followed by feedback interviews analogous to the previous ones. We intentionally decided on an iterative approach as this makes the specification of information needs more agile and, in our experience, stakeholders without any background in data science feel less overwhelmed by this task as they can incrementally learn and get an idea of what is possible.

After implementing six individual processes (one for each case) to monitor stakeholder-specific information needs and support their decision making, we analyzed the empirical data collected during the collaboration with the stakeholders. Specifically, we documented and compared 1) the stakeholders’ use cases and information needs; 2) the data sources used during the implementation; 3) the types of analyses and methods used during implementation; and 4) the individual procedures and steps to get to the stakeholder-targeted monitoring- and decision-making frameworks. As
a result, we derive a generic model that 1) provides an approach for utilizing low level metrics to incrementally get from monitoring basic information needs to generating continuous, strategic knowledge supporting decision making; 2) is adaptable to various stakeholders involved in the creation and distribution of software-intensive products; 3) and increases the efficiency of measurements by fostering the reuse of knowledge across stakeholders.

5.4 Related Work

This section provides an overview of existing research in the field of software analytics and data-driven measurement approaches targeting different roles involved in all company functions. In addition to that, we lay out the design rationales for the derived processes and model.

Oftentimes, existing approaches focus on supporting or answering questions for one specific stakeholder. In [Cito, 2016], for instance, runtime information is presented to software developers while writing new code in order to support their decision making. This is achieved by mapping operational data to source code artifacts which are then annotated with useful information (e.g. execution times) [Cito, 2016].

Buse and Zimmermann [Buse and Zimmermann, 2012] found that information needs of software developers are well-researched and understood. This is quite the contrary for other stakeholders such as (project) managers [Buse and Zimmermann, 2012]. It is critical to recognize that each stakeholder has individual needs and interests that need to be focused on. Augustine et al. [Augustine et al., 2017] implemented a software analytics solution that addresses multiple types of stakeholders (e.g. software architects or product owners) by measuring and visualizing a defined set of metrics. Their focus is very much on the automation, visualization, and deployment of these measurements, while our study centralizes more around a strategic concatenation of multiple analyses to generate stakeholder-targeted, in-depth knowledge and insights.

There exist different types of analytics that organizations can apply to gain insights on their business [Nayebi et al., 2015], [Saggi and Jain, 2018]. Depending on the variables used during the operations, measurements can either be direct (one variable) or indirect (multiple variables) [Kaner et al., 2004]. The operational level in our model is mainly designed for descriptive analyses using direct measurements, while the tactical level includes diagnostic and predictive techniques using indirect measurements, and the strategic level utilizes predictive and prescriptive methods with indirect measurements.

A very common method to measure the impact of new changes in software engineering is A/B experimentation [Kohavi and Thomke, 2017], [Fabijan et al., 2017a]. Two versions, A (control) and B (treatment), of the same product are deployed and the users get randomly assigned to one of the versions. Previously defined key metrics ("Overall Evaluation Criteria"), such as sales or click rates, can then be compared [Kohavi and Thomke, 2017]. This allows the identification of the product or feature value, and the changes that contribute to it [Fabijan et al., 2017a]. However, defining the overall evaluation criteria as well as aligning business key performance indica-
tors (KPIs) and team level metrics have been identified as key challenges with A/B experimentation [Olsson et al., 2017], [Fabijan et al., 2017b]. Specifying concrete information needs was also a major challenge for the participants in our study. Our model is, therefore, based on multiple feedback cycles, allowing step-wise definitions and adaptations of measurements.

The Goal Question Metric approach [Caldiera and Rombach, 1994], [Basili, 1992] is a common practice to measure a set of predefined goals. First, an overarching goal needs to be defined (e.g. related to products, processes or resources). Each goal is characterized by a set of questions while each question is associated with one or more metrics [Caldiera and Rombach, 1994]. Basili claims that "measurements must be defined in a top down fashion, bottom-up approach won’t work" [Basili, 1992]. In practice, however, the definition of high-level measurement goals can be quite challenging [Berander and Jönsson, 2006]. Indeed the participants of our study had a very difficult time specifying strategic questions or information needs. Moreover, two of our researchers found in a previous study on adopting data-driven development practices that it is important to keep things simple at the beginning, for instance by focusing on a small subset at first and scaling up at a later point in time [Olsson and Bosch, 2019]. Similarly, the approach of our model is designed in an iterative and step-wise manner, for instance by focusing on a subset of data sources for each stakeholder and lifting up the complexity of analyses step-by-step.

According to [Dam et al., 2018] and [Buse and Zimmermann, 2012] it is crucial for humans to understand the reasons behind decisions made by software analytics tools in order to trust their results. To achieve so-called explainability, the entire analytics process should be either transparent and understandable or complemented by an explicit explanation [Dam et al., 2018]. Moreover, it has proven to be beneficial to run multiple, layered analyses in order to provide various levels of detail and increase the usefulness of information. On top of that, information can even become an actual insight by providing stakeholders with a further understanding (e.g. why...? how...?) [Buse and Zimmermann, 2012]. We made very similar observations during our study. As a result, our approach is based on layers to 1) incrementally increase the degree of knowledge complexity (from operational to tactical so strategic), while at the same time 2) simplifying the explanation process by drilling it down to smaller chunks.

5.5 Implementation of Monitoring- and Decision- Making Frameworks

In our previous work [Figalist et al., 2019a] we introduced a generic model for continuous, real-time and strategic business insights based on low level metrics, specifically targeting product management. The model is based on three levels: operational, tactical, and strategic [Anthony, 1965]. These levels are a common dimension in the field of planning and decision-making in agile software development [Moe et al., 2012]. In recent years, the prominence of data-driven approaches in software engineering increased tremendously. Data emerges from many different sources throughout the entire software engineering process and a variety of analysis techniques is available to
basically anyone. Product management is by far not the only stakeholder benefiting from such analyses. Many other stakeholders could also benefit from jumping on the bandwagon to gain valuable, real-time insights. Therefore, our study focuses on four different types of stakeholders involved in the creation and distribution of software-intensive products.

The following subsections provide an overview of the individual monitoring- and decision-making frameworks implemented for different stakeholders, specifically for product managers (Cases A, C, and E), operation engineers (Case D), software architects (Case B), and sales (Case F). Furthermore, we use the observations made during the implementations of Cases A-F to derive a holistic model that enables continuous measurements of stakeholder-targeted information needs as well as the generation of strategic knowledge and insights throughout the entire software engineering process.

5.5.1 Case A: Product Management of Healthcare Platform

The implementation of Case A is described in the following paragraphs. Additionally, Figure 5-2 provides an overview of the step-wise approach. Each of the three columns in the figure represents one iteration (step I–III) of the case. The type of information needs or insights specified by the stakeholder evolves from operational, to tactical, to strategic and is summarized in the Insights-row of each iteration. The input data used in the respective iterations is listed in the boxes below the insights. In addition to that, the transformation of input data, to data analysis results, to knowledge extraction is outlined by showing (exemplary) results of the analysis as well as the knowledge gained from the analysis (bottom boxes) in each iteration.

During the initial stakeholder interview, the product manager explains that he is not entirely aware of how the application is used by its users. As a result, he first asks for the current usage rates of the application as well as information on the sessions during which the application was used.

**Step I: What are the current usage rates of the platform’s application?**

The input data of step I comprises many different and orthogonal low level metrics related to the usage of the system. To address the product manager’s information needs, we perform basic aggregation techniques to provide a certain understanding of the data. Specifically, we investigate the following metrics related to application usage: number of page views, maximum, minimum and average duration on each page, number of navigation paths, length of navigation paths etc. Next, we examine how often users visited a certain page in comparison to other pages, how much time users spent on each page, the relationship between pages (e.g. how many times users navigate from one page to another), and how often certain functionalities are used. All results are visualized in graphs or charts, like the example shown in the operational column of Figure 5-2 which presents the average duration users spent on each page.

These aggregations and their respective visualizations aim at answering basic questions on a rather operational level and serve as the groundwork that all further steps
What are the current usage rates of the applications?

- Duration on page
- Number of page views
- Number of chosen paths
- Length of chosen paths

Which behavioral patterns exist in regard to application usage?

- Important pages / filters / features for customers
- Rarely used features
- Navigation paths

How can the application’s websites be restructured to optimize usability?

- Important pages / filters / features for customers
- Rarely used features
- Probability of certain navigation paths
- Network navigation paths
- Session type classification

Figure 5-2: Implementation of Case A

are built upon. After presenting the results to the product manager, he got a first understanding of how the application is used. In more detail, the product manager can now derive the basic navigation paths users usually take while using the application. Additionally, by knowing the frequency of the page views on different levels within the website structure as well as the duration the user spent on each of the pages, he is able to determine how important certain pages or features are for the customer. Building upon this, a new and more complex information need is specified in collaboration with the product manager. He is now interested in behavioral patterns that exist in regard to application usage.

Step II: Which behavioral patterns exist in regard to application usage?

In order to address the more complex information need, we enrich the already existing input data by the knowledge gained in the previous step. We specifically focus on the navigation paths and patterns within the page views. For one, we construct a network graph that visualizes the page views as well as their attributes (number of calls and maximum, minimum, average duration) as nodes and their relationships (navigation paths) as edges.

In order to being able to simulate or predict user behavior, we use a Markov chain
[Gagniuc, 2017] to statistically model the navigation flows and their probabilities. This is a well-known method for analyzing and predicting navigation paths on the web by modeling past URL request patterns [Sarukkai, 2000]. Each web page is considered a state and Markov chains model the transition probability from one state to another while taking the dependencies of the previous state into consideration [Sarukkai, 2000]. The result of the Markov chain is visualized in the tactical column of Figure 5-2. Whenever a user visits a certain page of the platform (node) the Markov chain is applied to predict the next steps of that user based on the probabilities of adjacent paths. These transition probabilities are resembled by the width of the respective edges.

In addition to that, we analyze the page views across all sessions. Specifically, we cluster all sessions by similarity of page views within the sessions using a hierarchical clustering technique [Ward Jr, 1963]. We selected this technique as it has proven to be well-suited for document classification problems [Zhao et al., 2005] which is very similar to our problem space since each session can be considered as a document consisting of multiple terms (page views). During (agglomerative) hierarchical clustering a tree is built from bottom to top: Initially, each object (session or document) is assigned to its own cluster (leaves). By calculating the similarity between clusters, they are iteratively merged into larger clusters until only one overarching cluster remains (root) [Zhao et al., 2005]. To identify the final clusters, the tree is cut at the level of hierarchy that represents the predefined number of clusters (e.g., $k=10$). In our case, each cluster can then be considered a session type (e.g., user setting, technical settings, usage of application $x$) containing pages that are usually visited within that type of session. For example, if users mainly navigate through the technical settings of the application within one session, this session would be assigned to the cluster technical settings.

As a result, the product manager gains knowledge on the behavioral patterns of his application’s users, such as the session types or the probabilities of choosing certain paths, including a prediction model for the next steps taken by a user. By implementing this step in real-time, the product manager can even observe dynamic changes in user behavior, for instance after a new deployment, in order to detect or react to occurring abnormalities.

After getting a more in-depth understanding of how the application is used by its users, the product manager is interested in recommendations how to act on the newly gained insights. However, he finds it difficult to come up with a concrete type of recommendation. As a result, we propose to look into potential optimizations of the application’s web pages based on the usage patterns.

**Step III: How can the application’s websites be restructured to optimize usability?** Analogously to the previous step, we add the knowledge gained in *step II* to the existing pool of input data in order to answer the product manager’s most strategic question on optimization potentials. In this case, we follow the approach proposed by Singh and Kaur [Singh and Kaur, 2014]. They apply an ant colony based algorithm in order to optimize website structures. Ant colony optimization,
inspired by the behavior of real ants and their deposit of pheromones, aims at finding solutions for optimization problems, such as the shortest path for the well-studied Traveling Salesman Problem [Dorigo et al., 2006]. In our case, we aim at finding the shortest navigation paths required to achieve a specific goal in the application. The Markov chain of step II as well as the page view sequence of each session is used for the initialization of the ant colony based algorithm. During the execution, artificial ants move on a graph that represents the pages of the application. While doing so, the ants deposit pheromones on the graph’s edges which increases the probability of other ants choosing that edge. Since ants choosing shorter paths reach their goal and return from it more quickly, the pheromone level on these edges will increase faster as compared to others, leading to more ants choosing the shorter path. After multiple iterations the shortest paths are clearly visible. The pheromone level on each edge resembles the importance of a direct link between two web pages. Ultimately, this results in concrete suggestions for the product manager how to adapt the website structures in order to improve the navigation flows of the application. This is visualized in the strategic column of Figure 5-2. In theory, by iteratively repeating the three steps presented in this case, the product manager can get immediate feedback on how the optimized website structures, or any changes related to the measured metrics, influence user behavior (steps I and II), and if any further actions might be recommended (step III).

5.5.2 Case B: Software Architect of Industrial Platform

The implementation of Case B is described in detail in the following paragraphs. Figure 5-3 gives a first overview of the approach. The elements of the figure as well as their positioning are analogous to the implementation figure presented for the previous case (see Section 5.5.1 for a detailed description).

The software architects’ goal is to refactor and optimize the platform’s existing API design. Therefore, their initial information need is related to the existing design and how it is used by external developers.

**Step I: What are the current usage rates of the platform’s APIs?** As primary data sources we, therefore, select the API’s dependency data as well as static code metrics. In a first step, we aggregate usage rates of APIs by different applications (see operational column of Figure 5-3). This results in an overview of frequently and rarely used APIs as well as a mapping of APIs and applications that use respective APIs.

During the feedback interview, the software architects get a good idea of how the APIs are used individually. Consequently, they specify a new information need on the relationships between APIs, current usage patterns and the resulting complexity of the API design for its users.

**Step II: How complex is the current API design for its users?** We use the concrete mapping between application classes and platform APIs to generate a hier-
What are the current usage rates of the platform’s APIs?

- Platform API (method level)
- Number of methods per interface
- Number of interfaces used by an application

Basic statistics:

![Platform API Usage Rates]

- Frequently used APIs
- Rarely used APIs
- Mapping: Application ↔ APIs

**Tactical**

How complex is the current API design for its users?

- # of methods per interface
- # of interfaces used by an application
- Frequently/rarely used APIs
- Mapping: Application ↔ APIs

**Strategic**

How to refactor the current API design in order to decrease its complexity?

- # of methods per interface
- # of interfaces used by an application
- Mapping: Application ↔ APIs
- Interfaces / methods commonly used together

**Recommendation system for refactoring of API design:**

- SubA.CompA.API5 SubA.CompB.API3
- SubA.CompA.API1 SubA.CompB.API4
- SubA.CompB.API2 SubA.CompB.API2
- SubC.CompA.API2 SubC.CompA.API1

Figure 5-3: Implementation of Case B

Architectural clustering of the data set in order to determine which APIs are commonly used in combination with certain other APIs (see tactical column of Figure 5-3). A short description of how hierarchical clustering techniques work is provided in Section 5.5.1.

In addition to that, we generate association rules of APIs using the FPGrowth algorithm [Han et al., 2000]. Association rules are expressions (e.g., \( X \Rightarrow Y \)) describing an association between sets \( \{X, Y\} \) of one or more items [Agrawal et al., 1996]. This association implies that "transactions of the database which contain X tend to contain Y" [Agrawal et al., 1996]. In our case an example would be that "97% of applications that use SubsystemA.ComponentB.API1 and SubsystemA.ComponentB.API3 also use SubsystemC.ComponentA.API2."

These usage patterns indicate how complex the usage of platform APIs is for application developers. For instance, if the number of components (each component consists of multiple APIs) required to serve a specific use case is high, the developer needs to acquire a more extensive understanding of the platform itself which leads to a higher degree of complexity.

After presenting the results as part of the feedback interview, we received positive feedback from the software architects since the clusters as well as the association rules...
help them in understanding which combinations of APIs constitute common use cases of external developers. By evaluating their characteristics (e.g., number of different components within one cluster or association rule), the software architects are able to determine the complexity of using the platform’s APIs to achieve a specific use case. As their ultimate goal is to refactor the API design, the software architects specify recommendations and suggestions for refactoring as their next high-level information need.

**Step III: How to refactor the current API design in order to decrease its complexity?** The previously detected patterns can serve as a basis for refactoring the API design based on the actual usage of the APIs. We pursue two objectives by refactoring the design: 1) Optimization of maintainability for the provider; as well as 2) optimization of usability for external developers.

In this specific case, we use a genetic algorithm [Holland, 1992] to generate various solutions for a potential refactoring of the API design. This approach is inspired by Darwin’s theory of natural selection. During the initialization a pool of possible solutions (=population) is generated. In our case, we generate completely random as well as biased solutions. The latter are based on the clustering results obtained in step II. The precondition for the generated solutions is that all API methods of the original API design must also occur in the new design. A key element of genetic algorithms is the fitness function that evaluates the quality of all (initial and generated) solutions. In the fitness function we aim at quantifying the trade-off between quality code metrics (e.g., number of methods per API) and the average number of APIs and methods required to implement a use case. In multiple iterations the genetic algorithm generates alternative API designs based on the current population. Each iteration consists of the following steps [Sivanandam and Deepa, 2008]: 1) Selection of the $n$ best candidates found in the current population; 2) generation of new solutions based on the selected best solutions (crossover); 3) generation of new random solutions (to avoid local maxima); 4) calculation of fitness score; and 5) evaluation whether one of the solutions meets a predefined stopping criteria (e.g., threshold or no significant improvement of fitness score for $n$ iterations). Whenever a generated design fulfills the requirements of the fitness function, it is presented as a recommendation to the software architect. The strategic column of Figure 5-3 presents an exemplary recommendation to merge a subset of methods from two interfaces into one.

During the feedback interview, the software architects perceive this as helpful input that they can use as a starting point. However, since not everything can be modeled as a fitness function, there is still some manual work to do by the software architects.

**5.5.3 Case C: Product Management of Remote Connection Service Provider**

The implementation of Case C is described in the following subsections. Figure 5-4 provides an overview of the procedure. The elements of the figure as well as their positioning are analogous to the implementation figures presented for the previous
cases (see Section 5.5.1 for a detailed description).

In order to get a first understanding of how the platform is currently used, the stakeholders’ initial information needs are related to the current execution rates of the platform’s operations.

**Step I: What are the current execution rates of the platform’s operations?**

The input data for Case C consists of audit logs of the platform containing every operation executed by users over a period of two months. Each entry in the audit logs contains the operation ID, operation specific info (e.g. device IDs), a timestamp, and the user ID. Quite similar to the previous cases, we initially perform some basic data aggregations. We investigate the number of executions for each operation in order to determine which operations are common or frequent and, on the other hand, which operations are rather rare (see operational column of Figure 5-4). By grouping the operations by timestamps and user IDs we are able to reconstruct the order of operation calls.

After providing a first overview of the operation execution rates, the product managers get a first understanding of how intensively certain functionalities are used by its users.
For the upcoming iteration the product managers are interested in common operation sequences that users perform on a regular basis.

**Step II: Which sequences of operations are frequently executed by users?** Building upon the knowledge of frequent single operations in *step I*, we now aim at analyzing the frequency of operation sequences in *step II*. We combine the operation IDs as well as the order of operations to recognize common sequence patterns (see tactical column of Figure 5-3). This supports the product managers’ understanding of how their system is actually used and which operations are usually executed in combination with certain other operations as part of a sequence. In order to reduce the number of operation users need to execute to fulfill their use case, the next, more high-level information need is related to smart consolidations containing common operation sequences.

**Step III: Which operations can be consolidated into one to optimize the platform’s usability?** The final goal is to utilize recurring patterns in order to generate recommendations to simplify the usage of the platform. In a first step, we use the operations ID as well as the operation specific info to identify sequences of user role assignments to the same user. Next, we compare common combinations of user roles in order to generate recommendations on which roles can be consolidated into an overarching role comprising several sub-roles (see strategic column of Figure 5-4). Ultimately, this enables product managers to improve the offering of their product by reducing the number of operations for the users. By iteratively executing this procedure, product managers are able to monitor how the proposed changes affect the operation execution rates as well as the behavioral usage patterns.

### 5.5.4 Case D: Operations Engineer of Healthcare Platform

The implementation of *Case D* is described in the following paragraphs. Additionally, Figure 5-5 provides an overview of the step-wise approach. The elements of the figure as well as their positioning are analogous to the implementation figures presented for the previous cases (see Section 5.5.1 for a detailed description).

Since the operations engineers of the platform want to ensure the availability and reliability of the platform and its applications, they need to be aware of any incidents (service unavailability) that currently or recently occurred at customer sites. Therefore, we start by looking at the frequency of incidents in different locations and timeframes.

**Step I: How frequently do incidents occur in different locations and timeframes?** In order to address information needs of operation engineers, we initially focus on the incident data collected during the runtime of the product. We investigate the number of incidents occurred in different locations of deployment. In addition to regional information, temporal constraints are also taken into consideration (e.g.
How frequently do incidents occur in different locations and timeframes?

- Receiver downtime
- Receiver uptime
- Location data

Which correlations exist between incidents and the usage of the platform?

- Incident frequencies
- Incident locations
- Usage data of customers

Which type of incident is going to occur when?

- Usage data of customers
- Incident frequencies
- Incident locations
- Relationships between incidents and usage behavior or other factors

---

Figure 5-5: Implementation of Case D

The average number of incidents per day or number of incidents that occurred since the latest release has been deployed. As an example of the results, the number of incidents per day is visualized in the operational column of Figure 5-5.

As a result, information on incident frequency and incident locations are provided to the operation engineers. In the next step, we aim at identifying correlations between incidents and the usage of the platform.

**Step II: Which correlations exist between incidents and the usage of the platform?** While there is an easy explanation for some of the incidents’ root causes, others only occur in certain situations caused, for instance, by bottlenecks within the system. In these cases it is extremely critical to know and understand the context in which incidents occur. We aim at identifying potential relations between incidents and usage behavior of customers by calculating the Pearson correlation coefficients [Freedman et al., 1978]. Correlation coefficients can vary between 1 (strong positive correlation), 0 (no correlation), and -1 (strong negative correlation). A positive correlation between two variables $X$ and $Y$ exists if $X$ increases and $Y$ increases as well. Whereas a negative correlation is given if $X$ increases while $Y$ decreases. Even though correlation does not necessarily equal causality, it can still make sense to use...
it as a starting point for the identification of potential bottlenecks within a system, for example if a lot of users use a certain features at the same time, some of them will fail to get connected to the server. An extract of correlation analysis' results is presented in the tactical column of Figure 5-5.

Whenever an incident is reported, the operations engineers can take the correlations of incidents with other variables (e.g. Feature X) as a starting point to take a targeted approach at identifying the root cause and then forwarding it to the responsible team. In order to act timely on incidents or even take preventive actions before incidents occur, the next, more strategic information need is related to identifying and predicting which type of incident is going to occur when.

**Step III: Which type of incident is going to occur when?** By applying time series analysis to the data set, also temporal influencing factors can be taken into consideration. Operations engineers will then be able to identify three types of incidents: 1) Incidents that regularly occur during a specific timeframe (e.g. every time after a new release); 2) incidents that occur whenever a certain usage behavior can be observed (e.g. high usage rates of CPU-intensive features); and 3) incidents that regularly occur during a specific timeframe due to usage behavior (e.g. many customers upload their reports on Friday afternoons which will cause an overload on the server). Furthermore, the knowledge of existing bottlenecks can be used to implement a prediction model to predict upcoming incidents based on usage and time series data. For example, the previous correlation analysis of step II indicates a correlation between incidents and the usage rates of Feature X. Whenever the usage rate of Feature X increases to a critical level, an alert is sent to the operations engineers (see strategic column of Figure 5-5).

Consequently, the operations engineers are then aware of situations and also the frequency of situations in which incidents occur. Based on this, they can make informed decisions on forwarding incidents to responsible teams or taking and prioritizing potential countermeasures themselves.

### 5.5.5 Case E: Product Management of Industrial Platform

The implementation of Case E is described in detail in the following paragraphs. Figure 5-6 gives a first overview of the approach. The elements of the figure as well as their positioning are analogous to the implementation figures presented for the previous cases (see Section 5.5.1 for a detailed description).

During the initial stakeholder interview, the product managers of the platform are interested in how the platform’s APIs are currently used in order to better understand the needs of the platform’s external developers.

**Step I: What are the current usage rates of the platform’s APIs?** The initial input data consists of the usage dependencies between applications and platform APIs. Furthermore, each application identifier and each platform API identifier comprises an application namespace or a platform namespace respectively. Each
namespace can either contain several application classes or several platform APIs and methods. We extract the mappings between application classes and platform API methods out of the data set and perform some basic aggregations in order to generate some first insights which APIs are used very frequently, on a regular basis, or only rarely (see operational column of Figure 5-6). This provides a first hint which APIs are important to external developers and, therefore, valuable for the platform. After presenting the results to the product managers, they are interested in platform APIs that are used in combination with other platform APIs.

**Step II: Which platform APIs are commonly used in combination?** On a more complex level, the product management is interested in common API usage patterns. In order to get a deeper understanding of how the platform APIs are currently used by application developers, we reuse the results of the hierarchical clustering and association rule mining conducted in Case B to identify APIs and interface methods that are commonly used in combination with certain other APIs (see tactical column of Figure 5-6).

Over the course of several years, the numerous, newly added APIs have made the
usage of platform APIs complex for external application developers. After taking a
closer look at the association rules and the clustering results, the product managers
gain a better understanding of common API combinations and dependencies between
applications and APIs. Based on these insights, the product managers would like to
simplify the usage of APIs for their external developers.

Step III: How to simplify the API usage for its users? Lastly, we use the
clustering results obtained in step II as the input data for step III. We train a model
based on the clustering which takes the currently APIs of application developers as
input and then predicts the cluster that this application and its corresponding plat-
form APIs belong to. By comparing the platform APIs of the respective cluster to the
actual APIs used by the application, the platform provider can improve its offerings
by providing tailored recommendations to the app providers which other platform
APIs might be of interest to them. This recommendation process is visualized in
the strategic column of Figure 5-6. By continuously executing steps I – III in an
automated pipeline, the recommendations get more precise with each iteration, and
the platform provider can observe how the utilization of platform APIs evolves over
time.

After presenting the results to the product managers during the final interview, they
provided positive feedback and consider the recommendation system a valuable asset.

5.5.6 Case F: Sales Team of Healthcare Platform

The implementation of Case F is described in the following subsections. Figure 5-7
provides an overview of the procedure. The elements of the figure as well as their
positioning are analogous to the implementation figures presented for the previous
cases (see Section 5.5.1 for a detailed description).

The platform’s license model includes a basic license, which is free whenever one
of the complementing hardware devices is purchased, as well as a premium license
that offers additional functionalities and prioritized support of customers. Before
purchasing the premium version of the product, each customer can choose to enter
a trial phase during which the premium version can be tested for free over a period
of three months. Since premium licenses can also be purchased for each application
individually, there are a lot of different license types and combinations of these types
offered by the platform provider.

Therefore, the sales representative team would first like to get an overview of which
customers purchased what kind of license types.

Step I: What are the current sales rates of different license types? The
primary data sources of Case F consist of the platform’s and applications’ license
data records as well as the master data records (e.g. registration dates).

In order to provide a first understanding of the data and potential information and
insights hold by it, we start by aggregating the number of types each license type has
been purchased by customers in the past. We also look into regional and temporal
### Insights

What are the current sales rates of different license types?
- Number of basic licenses
- Number of premium licenses

Basic statistics:

<table>
<thead>
<tr>
<th>Licenses</th>
<th>Basic</th>
<th>Premium</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>700</td>
<td>500</td>
</tr>
<tr>
<td>100</td>
<td>600</td>
<td>400</td>
</tr>
<tr>
<td>200</td>
<td>500</td>
<td>300</td>
</tr>
<tr>
<td>300</td>
<td>400</td>
<td>200</td>
</tr>
<tr>
<td>400</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>500</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>600</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Data

- Number of sales per year/region...
- Ratio of basic and premium customers

### Operational

What is the difference in behavior of "basic" and "premium" customers?
- Sessions per customer
- Purchasing behavior
- Page views clustering of Case A

### Tactical

Session type classification:

- app/events
- app/analysis
- app/customers
- Technical Settings
- settings/device
- settings/dataAccess
- settings/categories
- settings/roleManagement
- settings/fleetManagement
- usage of application y

### Strategic

Customer churn prediction model:
- CustomerID=118, churn=0, Predicted=[False]
- CustomerID=537, churn=1, Predicted=[True]
- CustomerID=90, churn=0, Predicted=[False]
- CustomerID=147, churn=1, Predicted=[True]
- CustomerID=581, churn=1, Predicted=[True]
- CustomerID=32, churn=0, Predicted=[False]

### Knowledge

- Customer churn predictions
- Potential factors for churn based on feature weights

### Figure 5-7: Implementation of Case F

The data indicates that the majority of customers sticks with the basic license instead of paying for the premium version. During the feedback interview, the sales representative would like to know why this happens and, therefore, asks us to investigate whether there is any difference in behavior between basic and premium customers.

**Step II: What is the difference in behavior of "basic" and "premium" customers?** As a result to step I, we continue by looking into and comparing the usage behavior of basic and premium customers. We analyze the visited pages by sessions and end-users of each customers. In more detail, we reuse the results of the hierarchical clustering obtained in Case A to identify pages that are commonly visited within the same session. The identified clusters resemble session types that are frequently carried out by the system’s users. For instance, we identified the sessions user settings, technical settings, and usage of application y during which the majority of page views are related to the respective topics. An exemplary mapping between page views and session types is visualized in the tactical column of Figure 5-7. As
a result, we are able to compare the usage behavior, specifically based on single page views as well as the more comprehensive session types, of basic and premium customers. The results show that premium customers are usually quite active, while the activity range of basic customers is much larger. After discussing the results with the sales representative, he specifies the prediction of the customers’ purchase behavior after the trial phase as his next, more high-level information need.

Step III: Which customers are going to downgrade to a "basic" license? As the overarching goal of the stakeholder is to increase the number of sales, it is of high relevance to identify and predict customers who are potentially going to stop paying for a product in the near future, also known as customer churn prediction [Mozer et al., 2000]. Therefore, we extract and label the usage behavior of customers, including the previously identified session types, throughout different phases of a customer’s lifecycle in order to train a prediction model. In more detail, the extracted input data set is preprocessed and split up into a training data set (80%) and a test data set (20%). We use the Sequential model of the Keras package [Gulli and Pal, 2017] to build a neural network using 25 selected input features, one target variable and two hidden layers with twelve nodes each. The parameters are initially set based on the shape of the input data (e.g. 1:2 ratio of nodes and input features) and are then optimized using a trial and error approach. The model is trained on the previously extracted training data set. Finally, we evaluate our model by predicting the output variable of the test data set, achieving an accuracy of 88% with a precision of 0.89 and recall of 0.91 (a more thorough description of the approach is provided in [Figalist et al., 2019b]). An extract of the model’s prediction results is shown in the strategic column of Figure 5-7.

The results of the prediction model’s outcome continuously guide sales stakeholders to proactively contact specific customers in order to prevent them from churning. During the final feedback interview, the sales representative provides positive feedback and states that he would like to include this analysis in the existing sales dashboards.

5.6 Results: Generic process and model for efficient instantiations of monitoring- and decision-making frameworks

This section describes the findings observed during the implementation of the previously described cases. For one, we describe the generic process as well as the kind of data and methods that are most relevant to individual stakeholders we address in our study. Furthermore, we present the data and knowledge flow of each stakeholder across different levels (operational, tactical, and strategic) and highlight overlaps and reuse potentials. Ultimately, we use these insights to derive a generic model.
5.6.1 Primary data sources and methods

One observation that stuck out during the implementation of the individual cases is the relevance of different types of data to respective stakeholders. Especially in the operational level the generated information is mainly based on one or two different sources of data. We use this knowledge to identify so called primary data sources for each stakeholder. Primary data sources are the main data sources required to generate stakeholder-specific knowledge. During the course of our analysis we utilize the following data sources: dependency data, static API data, incident data, audit logs, usage data, master data records, and license data. We compare the list of available data sources to the ones actually used for each of the stakeholders. Based on their underlying interests and goals, we are able to assign subsets of the data sources as primary data sources for each stakeholder.

The software architects of Case B, for instance, are mainly concerned with a robust and maintainable architecture of the system. Therefore, their primary data sources are the dependency as well as the static API data of the system. The operations engineers (Case D) want to ensure the availability and reliability of a system during runtime. For that reason, we identified the incident data as the primary data source for this stakeholder. The product managers of our study (Cases A, C, and E) are mainly interested in how customers are using their product. This is primarily resembled by the audit logs, usage data, as well as master data. Lastly, the sales staff (Case F) focuses on selling the product and, in the best case, increasing the number of sales. Hence, the primary data sources for this stakeholder are the license as well as master data records.

The primary data sources are not set in stone and can be iteratively adapted to fit the stakeholders’ information needs. The purpose of assigning primary data sources to the stakeholders at hand is to narrow down the problem space in order to generate results faster and more efficiently.

Moreover, we identified four method categories used to generate insights across all stakeholders. Each measurement procedure started out with some basic statistics and aggregations to get a first understanding of the current situation. In a next step, more complex methods are applied, mostly to identify patterns within the data. Based on the detected patterns, we are able to implement more complex prediction models or recommendation systems.

Figure 5-8 visualizes the primary data sources for each of the stakeholders in our study and highlights the knowledge and insights generated by each of the methods. The available data sources are listed at the bottom of the figure and the table summarizes the stakeholder-targeted information across stakeholder and method types. The boxes on the left resemble the primary data sources we used during implementation for each type of stakeholder.

5.6.2 Data and knowledge flow across stakeholders and levels

Another observation made during the implementation phase is the opportunity to reuse different components across stakeholders. While each stakeholder naturally fol-
ows its own agenda, we stumbled into various situations in which we were able to reuse an analysis that was originally made for a different stakeholder. One example for this is the product management of Case E who wanted to understand how external developers are using the system’s APIs in order to provide customized recommendations to them. In this case we were able to reuse the API dependencies as well as the API usage patterns, originally aggregated and analyzed for the software architect in the operational and the tactical level (Case B). Moreover, we realized that especially in the strategic level stakeholders highly benefit from the tactical knowledge generated for other stakeholders as these types of insights are often influenced by multiple factors. The customer churn prediction model, for instance, also relies on usage as well incident patterns since the churning of customers is highly related to the customer’s usage and the products reliability. Analogously, the operations engineer was interested in bottlenecks that are responsible for certain incidents. This makes it inevitable to not include usage patterns in the analysis. Figure 5-9 gives an overview of the data and knowledge flow across levels and highlights which components were reused for additional stakeholders. Each of the colored boxes resembles stakeholder-targeted knowledge which is mapped to its respective stakeholder and complexity level. The arrows indicate which previously generated knowledge is used as input to generate additional and more complex stakeholder insights. This knowledge flow can be observed for individual stakeholders as well as across different types of stakeholders within the same product team.

5.6.3 Generic process for instantiating monitoring- and decision-making frameworks

By comparing the individual cases and implementations described above, we are able to derive a generic model guiding the process to systematically get from low level metrics to operational, tactical, and ultimately strategic knowledge. Figure 5-10 provides a detailed overview of the approach. It consists of three levels
Figure 5-9: Data and knowledge flow across stakeholders and levels

represented by the three columns in the figure: operational, tactical and strategic [Figalist et al., 2019a]. The purpose of these levels is to systematically process the data to incrementally lift up the complexity of knowledge generated in each level. There are standardized process steps defined to be passed through in each level. These process steps are represented by the three rows in the figure: the input data resembles the basis for each level, followed by a planning and execution phase in which the necessary data as well as the required operations (e.g. measurements, algorithms or techniques) are specified and executed. The results are then interpreted and processed to answer stakeholders’ information needs and generate actual knowledge. The knowledge itself already provides insights to stakeholders on different levels (operational, tactical, strategic). Additionally, the knowledge gained in previous levels is used as input data in the higher levels. In order to speed up value delivery, knowledge that was initially generated for other stakeholders within the same product team should be reused where it is appropriate. The same applies to methods and tools that can be used across stakeholders.

One key principle behind our approach is the strategic assembly of the individual levels in order to reach continuous and efficient generation of operational, tactical, and strategic knowledge. The operational level mostly includes the description of predefined KPIs or basic interests. Its main advantage is the capability to provide fast and immediate feedback of the system’s current behavior. Even unexpected behavior will most likely first show up on this level since it is closest to the actual data. The tactical level builds upon the operational level and aims at highlighting and monitoring patterns within the data. By combining multiple data sources, the
tactical level provides a deeper understanding of how certain behaviors occur and which patterns or relationships might exist. This enables stakeholders to comprehend actual behavior of data and to base and monitor their decisions on gained knowledge. Building upon this, the strategic level aims at analyzing the tactical knowledge in order to identify further suggestions, recommendations, explanations or predictions that can guide and support the stakeholders’ decision making in their concrete area of responsibility.

Another key aspect of this process is the sharing and reuse of methods and generated knowledge across stakeholders. This increases the transparency within product teams as well as the overall efficiency of the instantiations. In our study, for instance, we were able to reuse the API clustering originally implemented for the software architect of the industrial platform for the product manager of the same platform in order to generate customized recommendations for its users. Another example is the session type classification implemented for the product managers of the healthcare platform. We used this knowledge on the users’ behavior to train a churn prediction model for the sales team. Figure 5-11 gives an exemplary overview of how knowledge can be shared and reused across stakeholders. The colored boxes per stakeholder represent the individual instantiations of the previously presented generic process. In addition to that, the arrows indicate how the generated knowledge can be reused and shared across different types of stakeholders.

5.6.4 Model for efficient measurement and generation of stakeholder-specific knowledge

By combining the generic process for instantiating stakeholder-targeted monitoring-and decision-making frameworks with the newly gained knowledge about primary data sources as well as reuse capabilities, we are able to derive a generic model that
enables continuous and efficient measurements of operational, tactical and strategic knowledge for various stakeholders while being highly flexible in regards to defining stakeholder information needs as well as adaption of data collection. By focusing on primary data sources at the beginning and fostering the reuse of components later on during the analysis, our model can increase the efficiency of analyses and, thereby, create value to stakeholders very fast and early on.

Figure 5-12 presents the model in more detail. The primary data sources in the left box serve as a starting point within the continuous loop. Initially, the primary data sources for the stakeholder at hand are selected out of the pool of data sources. This way the risk of spending too much time on analyzing irrelevant data is minimized while the focus lies directly on the stakeholders’ interests. The selection of primary data sources can be adapted after each iteration and is, therefore, a self-optimizing process.

Furthermore, a central Reuse Repository stores and allows access to previous results that can be divided into three categories:

1) *Data & Knowledge* consisting of aggregated data, components, and knowledge generated during the analysis;

2) A *Toolbox* containing a collection of methods that can be used in different levels in order to achieve certain tasks (e.g. basic aggregation techniques, clustering techniques, calculation of correlation coefficients,...);

3) *Concepts* developed for either a specific type of stakeholder (e.g. product management) or a specific type of input data (e.g. static API data or dynamic usage data).
In most cases the sharing of Data & Knowledge will be limited to stakeholders within the same product team due to data privacy. However, the Toolbox and Concepts can be shared and reused within product teams as well as across product teams (product-external).

In a first definition phase, the stakeholders can define an initial set of operational information needs that they are interested in. Based on this, the selected input data is analyzed using appropriate methods selected out of the Toolbox within the Reuse Repository in order to generate the operational knowledge. In case a new method or technique is used, it is added to the list of tools. The Toolbox simplifies and speeds up the process of searching for appropriate methods suitable for a specific task. Existing Concepts provide guidance for analyses targeting specific stakeholders or specific types of data.

Moreover, the operational knowledge is stored in the central Data & Knowledge pool. The purpose of this pool is to join the knowledge generated in various levels for various stakeholders in order to 1) enable the reuse of components to increase efficiency; and 2) increase the transparency of analyses and knowledge across stakeholders. Next, the operational knowledge is presented to the stakeholders in order to collect their
feedback and jointly define information needs and potential insights of the tactical level. We specifically chose to not define all information needs at the very beginning because in our experience most of the stakeholders are overwhelmed by this task and have difficulties in describing what they would like to know. By iteratively discussing the results, the stakeholder can get a better idea of what is possible and can continuously drive and adapt the direction of generated knowledge.

After the tactical information needs are defined, the original input data can be enriched by any knowledge available in the Data & Knowledge pool before being analyzed by methods chosen out of the Toolbox. Analogously to the previous phase, the tactical knowledge is added to the Data & Knowledge pool and discussed with the stakeholder in order to define the more complex strategic information needs. Once again, all knowledge available in the Data & Knowledge pool can be used to enrich the existing data set that serves as input for generating strategic knowledge. The latter is also added to the Data & Knowledge pool before being discussed with the stakeholder. Based on the stakeholder feedback, the data collection can be adapted to fit the stakeholders’ needs. For instance, primary data sources can be reassigned in case the stakeholder wants to broaden or refine the scope of information needs or new data sources can be introduced to enable additional analyses.

This entire process can be repeated $n$ times for a various set of stakeholders. During each iteration further knowledge, tools, and concepts are added to the Reuse Repository. Finally, the results of our study show that this approach proved to be efficient in accelerating the instantiation process of monitoring- and decision-making frameworks for multiple stakeholders.

5.7 Conclusion

While data analytics is a very powerful tool to process data into miscellaneous types of information, insights, and knowledge, companies often struggle to make actual use of it. In previous studies, keeping track of collected data [Olsson and Bosch, 2013], sharing it appropriately [Fabijan et al., 2016], specifying information needs in a quantifiable way [Figalist et al., 2019a], and interpreting data [Buse and Zimmermann, 2012] have been identified as major challenges.

Our study, therefore, aims at investigating how to efficiently utilize data analytics to generate stakeholder-targeted information and knowledge. We use the results of six individual cases to derive a generic model that enables continuous and efficient monitoring of information needs and generation of knowledge and insights for various stakeholders in order to support decision making. The model fosters the sharing and reuse of knowledge and components, tools, and concepts to speed up the instantiation process. The process itself is based on a set of primary data sources and methods that we identify for each stakeholder during our analyses.

One threat to the internal validity of our study is the number of cases per stakeholder. While we worked with multiple stakeholders of the same type (e.g., operation engineers), they sometimes belong to the same product team and are, therefore, only considered as one case. Moreover, the number of cases for each product is not equally
distributed but varies between one and three.
The selected cases are based on unrelated and individually developed products. However, they all originate from a similar industrial business-to-business context which might impact the external validity of our study.
At this stage and since the study is based on voluntary participation, our findings are limited to four different types of stakeholders: product managers, software architects, operation engineers, and sales. However, other types of stakeholders (e.g. developers or customer service) could also be considered in future research. Another limitation of our study is the availability of data sources. During the course of this study, the platform providers gave us access to different data sources available for their respective products, including dependency data, static API data, incident data, audit logs, usage data, master data records, and license data. While we were limited to these data sources, in general there are many more data sources that could provide valuable insights (e.g. issue or bug tracking systems).
In future work we plan on extending the study to additional stakeholders and data sources as well as validating our approach in additional companies providing products in non-industrial contexts.
Chapter 6

Mining Customer Satisfaction on B2B Online Platforms using Service Quality and Web Usage Metrics


Abstract

In order to distinguish themselves from their competitors, software service providers constantly try to assess and improve customer satisfaction. However, measuring customer satisfaction in a continuous way is often time and cost intensive, or requires effort on the customer side. Especially in B2B contexts, a continuous assessment of customer satisfaction is difficult to achieve due to potential restrictions and complex provider-customer-end user setups. While concepts such as web usage mining enable software providers to get a deep understanding of how their products are used, its application to quantitatively measure customer satisfaction has not yet been studied in greater detail. For that reason, our study aims at combining existing knowledge on customer satisfaction, web usage mining, and B2B service characteristics to derive a model that enables an automated calculation of quantitative customer satisfaction scores. We apply web usage mining to validate these scores and to compare the usage behavior of satisfied and dissatisfied customers. This approach is based on domain-specific service quality and web usage metrics and is, therefore, suitable for continuous measurements without requiring active customer participation. The applicability of the model is validated by instantiating it in a real-world B2B online platform.
6.1 Introduction

With today’s technologies, it is easy for software vendors to provide their services fast and in an easily accessible way. Since this often results in an overflow of offerings for customers, service providers need to distinguish themselves in order to survive in such highly competitive markets.

Previous studies report empirical evidence for the impact of customer satisfaction on customer loyalty [Eklof et al., 2018], [McDougall and Levesque, 2000], repurchase intentions [Anderson and Sullivan, 1993], [Eggert and Ulaga, 2002], and on company profitability [Eklof et al., 2018].

The most common technique for assessing customer satisfaction are customer surveys [McColl-Kennedy and Schneider, 2000], [Peterson and Wilson, 1992]. However, the evaluation of such surveys is time consuming, the results are often skewed, and it is challenging to derive useful implications [Peterson and Wilson, 1992].

For the improvement of customer satisfaction, it is crucial to consider context-specific characteristics. Especially in business-to-business (B2B) markets the purchasing decision is quite rational [Patterson et al., 1996]. In addition to that, while the customers handle the purchase of a product, they are not the ones using it [Kandeil et al., 2014].

The end-users are typically an additional entity such as the customers’ employees. This complex setup often grouped with contractual regulations and restrictions make it difficult to assess customer satisfaction in B2B contexts, which results in the need for automated and quantitative measurement approaches.

While there are some indications that the application of web usage mining has a positive impact on customer satisfaction in business-to-customer (B2C) contexts [Zumstein, 2012], [Spiliopoulou and Pohle, 2001], it is not yet clear whether this also applies to B2B and, in addition to that, there is a lack of research on whether and how it can be used to quantitatively measure customer satisfaction.

Therefore, we aim at answering the following research question: How can customer satisfaction be automatically and quantitatively measured in B2B online service contexts?

To address this, we perform a literature review of approaches, principles and metrics for both customer satisfaction and web usage mining, as well as on B2B service characteristics. These results are then used to develop a model that uses measurable service quality metrics (SQMs) to calculate customer satisfaction scores. In parallel, we aggregate the usage behavior of end users on a customer level to examine the relationship between usage and service quality metrics. These insights are used to describe the usage behavior of different satisfaction classes and to confirm the appropriate classification of customers to their level of satisfaction.

Finally, the model’s applicability is validated by instantiating it in a real-world B2B online platform achieving a faster and more regular assessment of customer satisfaction in B2B contexts as compared to customer surveys alone.

The contribution of this paper is two-fold: First, we provide an approach to automatically and quantitatively measure customer satisfaction without requiring direct customer contact or efforts on the customer side. Second, we find empirical evidence for a relationship between perceived service quality and usage behavior allowing a
cross-validation of results obtained by the approaches. The remainder of the paper is structured as follows: Initially, a short overview of the research method and the study design is provided in Section 6.2, before presenting the results of the literature review in Section 6.3. Section 6.4 introduces the derivation of the model and Section 6.5 describes the instantiation of the model. Related work is presented in Section 6.6, before discussing the threats to validity in Section 6.7 and concluding in Section 6.8.

6.2 Research Method & Study Design

6.2.1 Research Process

An overview of the research process is provided in Figure 6-1. As our main research goal is to combine three already well-researched domains, we chose a deductive approach for our study. Deductive research approaches start by building hypotheses from existing theories which are then either confirmed or rejected based on observations [Runeson et al., 2012].

To provide a framework for our study, we started by conducting a literature review executing the following steps: 1) identification of research topics based on research question; 2) selection and quality assessment of studies; and 3) extraction of relevant information [Keele, 2007].

The research question contains several research topics that need to be set in context with each other. For one, we need to identify the basics and existing measurement methods of customer satisfaction. For another, the current state of the art in the area of web usage mining must be assessed. Lastly, domain-specific aspects of B2B service markets also need to be taken into consideration.

To achieve this, we select relevant research in each area individually and additionally in combination with each other in order to get an overview of the foundations as well as of related work that already combined two or more of the domains.

Based on the insights gained from the literature review, we deduced a model that leverages and combines principles of the three fields. Specifically, we used the knowledge on customer satisfaction and its influencing factors to derive SQMs based on
existing service quality measurement scales that can be calculated in a quantitative way.

In addition to that, we identified relevant web usage metrics in existing literature and followed the state of the art web usage mining process to extract these metrics on a user level. Moreover, we use the existing knowledge in the B2B area to aggregate user behavior on a customer level. Based on empirical evidence in the literature, we are able to link the usage metrics of a customer to its level of satisfaction. The applicability of the model is validated by instantiating it in a real-world B2B online platform. Further details on the case company and the instantiation process are provided in the following subsection.

6.2.2 Case Company

The case company is a platform provider established in the healthcare domain. The platform hosts several applications that can be purchased by healthcare providers. From a business model's perspective, the platform is considered a platform as a service (PaaS) and the applications as software as a service (SaaS). The company pursues a freemium pricing strategy for its applications, as the basic licenses are provided free of charge, while a premium license must be purchased for functionally enriched applications.

The platform provider granted us access to their web and application server logs, usage data, and (complaint) ticket data. We applied web usage mining to the logs and the usage data, while using the ticket data to calculate the SQMs. In addition to that, we conducted semi-structured interviews with the provider’s intermediaries between customers and development to evaluate and rank service quality dimensions extracted from literature. Finally, the outcome of the model was made available to the provider to support multiple stakeholders in their daily work.

6.3 Literature Review

As our study aims at combining knowledge on customer satisfaction, B2B service characteristics, and web usage mining, the following subsections provide an overview of the individual components based on existing literature.

6.3.1 Customer Satisfaction

To provide a foundation for our study, three aspects are explained in more detail in the following: 1) Definition, origin and consequences of customer satisfaction; 2) dimensions of service quality; and 3) measurement of customer satisfaction.

Definition, Origin and Consequences

Kotler et al. define satisfaction as "a person’s feelings of pleasure or disappointment resulting from comparing perceived performance or outcome in relation to his or her expectations" [Kotler, 2019].
One of the foundations of customer satisfaction (CSAT) research is the disconfirmation paradigm that was originated by Oliver [Oliver, 1980] and further developed by Parasuraman et al. [Parasuraman et al., 1988]. They describe service quality (SQ) as "discrepancy between consumers’ perceptions and expectations" [Parasuraman et al., 1988]. If the perceived performance meets (confirmation) or exceeds the expectations (positive disconfirmation), the customer is satisfied. Otherwise, it is dissatisfied (negative disconfirmation). As a result, service quality can be seen as both an antecedent [Cronin Jr et al., 2000], [Anderson and Sullivan, 1993], [McDougall and Levesque, 2000] and a consequence of customer satisfaction [Parasuraman et al., 1988].


To summarize the identified interrelationships, Figure 6-2 outlines the dependencies between terms in a simplified form.

Figure 6-2: Antecedents and consequences of customer satisfaction ([Oliver, 1980], [Parasuraman et al., 1988], [Anderson and Sullivan, 1993], [Eggert and Ulaga, 2002], [Eklof et al., 2018], [Kotler, 2019], [Bolton and Lemon, 1999], [Cronin Jr et al., 2000], [Cronin Jr and Taylor, 1992], [McDougall and Levesque, 2000], [Zeithaml et al., 1996])
### Table 6.1: Definition summary of the SaaS-QUAL factors [Benlian et al., 2011]

<table>
<thead>
<tr>
<th>Factor</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rapport</td>
<td>Providing knowledgeable, caring, and courteous support and individualized attention</td>
</tr>
<tr>
<td>Responsiveness</td>
<td>Ensuring the availability and performance of the SaaS-delivered application and the responsiveness of support staff</td>
</tr>
<tr>
<td>Reliability</td>
<td>Performing the promised services timely, dependably, and accurately</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Customer’s freedom to change contractual or functional/technical aspects in the relationship with SaaS vendor</td>
</tr>
<tr>
<td>Features</td>
<td>The degree of the key functionalities and design features of a SaaS application meets the business requirements of a customer</td>
</tr>
<tr>
<td>Security</td>
<td>Ensuring that regular (preventive) measures are taken to avoid unintentional data breaches or corruptions</td>
</tr>
</tbody>
</table>

#### Dimensions of Service Quality

In order to identify reasonable actions for enhancing customer satisfaction, there must be a clear understanding of what influences customer satisfaction in service contexts and an awareness of improvement potentials for those dimensions.

Existing literature has a very strong focus on measurement scales of *service quality* as opposed to *customer satisfaction*. However, due to their close interrelationship and the empirical evidence that service quality has a linear, positive impact on customer satisfaction, it is assumed that service quality dimensions also apply in the customer satisfaction domain.

In 1988 Parasuraman et al. [Parasuraman et al., 1988] developed the SERVQUAL instrument in order to measure service quality. As this scale was originally not intended for measuring the service quality of online services, it was extended and adapted by several researchers to fit the needs of software-intensive systems.

In the software as a service (SaaS) context, for instance, Benlian et al. [Benlian et al., 2011] extended and adjusted existing models to the SaaS-QUAL instrument. Apart from the familiar but adapted service quality dimensions rapport, responsiveness, reliability, and features, the factors security and flexibility were added. A more detailed explanation of the SaaS-QUAL dimensions is provided in Table 6.1.

In addition to that, Table 6.2 summarizes and compares five different *service quality measurement scales*. The choice of scale depends on the use-case context.

#### Measuring Customer Satisfaction

Most of the previously presented service quality scales measure customer satisfaction by conducting customer surveys, which are the most common evaluation method [McColl-Kennedy and Schneider, 2000], [Peterson and Wilson, 1992]. However, the results of these surveys are often negatively skewed while the majority of participants position themselves on the satisfied end of the scale. Therefore, differences between survey results are slight and managerial implications are difficult to deduce [Peterson and Wilson, 1992].

There exist two approaches to categorize *customer satisfaction* evaluation methods: 1) techniques can be either *subjective* or *objective*; and 2) be based on *direct* or *indirect* customer feedback [Grigoroudis and Siskos, 2009]. The latter is assessing whether the data used to evaluate customer satisfaction is coming directly from the
Table 6.2: Evolution and comparison of relevant service quality measurement scales ([Parasuraman et al., 1988], [Kettinger and Lee, 2005], [Parasuraman et al., 2005], [Swaid and Wigand, 2009], [Benlian et al., 2011])

customer or not. Objective methods are characterized by their objectiveness as they are not manipulated by subjective perceptions. As a result, they are not influenced by, but also do not include, perceived customer perspectives and as customer satisfaction consequences, they are time delayed and potentially neglect other relevant determinants. Subjective methods, on the other hand, are measuring psychological characteristics that are perceived differently from customer to customer [Grigoroudis and Siskos, 2009]. Table 6.3 maps common customer satisfaction evaluation methods along the two dimensions objectiveness of data and data origin.

Table 6.3: Customer satisfaction measurement methods

6.3.2 Characteristics of B2B Service Markets

To improve customer satisfaction, companies need to be aware of the specific markets and business models they are opting for, as different determinants exist for different products and services.

In their literature review, Homburg and Rudolph [Homburg and Rudolph, 2001] found
that satisfaction in B2C markets is mainly related to a single transaction, whereas in the B2B sector supplier-customer relationships are being promoted. They are described as "long-term oriented, enduring, and complex" [Homburg and Rudolph, 2001]. The customer is seen as an "active partner" rather than a "passive buyer" [Homburg and Rudolph, 2001].

The authors highlight that in B2B contexts the purchase decision is made by a buying team that often consists of cross-dimensional functions. In addition to that, purchase processes in organizations are more rational and systematic than those of individual customers [Patterson et al., 1996]. After a purchase is made, the product is typically not used by the customer but by an additional entity (e.g. the customer’s employees) [Kandeil et al., 2014]. Due to various communication channels between customer and end-users, the overall customer satisfaction depends on the perception of both entities which makes it very difficult to assess.

6.3.3 Web Usage Mining

Srivastava et al. define web usage mining as the "application of data mining techniques to discover usage patterns from web data, to understand and better serve the needs of web-based applications" [Srivastava et al., 2000]. By the application of web usage mining, knowledge about user profiles, behaviors, and interests can be derived and transformed into measures to ultimately improve websites and web applications [Liu, 2011].

The following subsections provide an overview of 1) the generic process of web usage mining; and 2) common web usage metrics.

Web Usage Mining Process

The overall process of web usage mining can be structured in three phases: 1) Data collection and preprocessing, 2) pattern discovery, and 3) pattern analysis [Liu, 2011]. In the first phase, web and application server logs are extracted, preprocessed and stored in a user transaction database while taking website and domain-specific knowledge into consideration. Next, usage mining techniques can be applied to the data stored in the transaction database which results in a set of usage patterns. These patterns can be analyzed in order to aggregate user models that characterize user behavior. Figure 6-3 provides a structured overview of the phases and their required steps.

Web Usage Metrics

Zumstein [Zumstein, 2012] presents an overview of potential metrics in the web and e-commerce context. To decide which measures to select, the objectives of the website and the maturity of the business model need to be taken into consideration. Table 6.4 presents a detailed overview of the introduced web metrics and their calculations.
6.4 Model Derivation

Based on the results of the literature review, we derive a model (see Figure 6-4) that connects principles of customer satisfaction and web usage mining while taking B2B specific characteristics into consideration. This enables a quantitative measurement of customer satisfaction in B2B contexts that does not require any active participation by the customers.

Initially, there are a number of steps that need to be taken in order to prepare the model for its application. This involves the extraction of relevant information in each web metric.

<table>
<thead>
<tr>
<th>Web metric</th>
<th>Calculations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page views</td>
<td>Average number of [unique] page views [per session] in a timeframe</td>
</tr>
<tr>
<td>Sessions</td>
<td>Average number of sessions in a timeframe</td>
</tr>
<tr>
<td>Users</td>
<td>Daily, weekly, monthly active users; new or returning users</td>
</tr>
<tr>
<td>Bounce rate</td>
<td></td>
</tr>
<tr>
<td>Page stickiness</td>
<td>[ \begin{align*} \text{sessions with a single page view} \end{align*} ]</td>
</tr>
<tr>
<td>Page view duration</td>
<td>[ \begin{align*} \text{sessions per unique visitor} \end{align*} ]</td>
</tr>
<tr>
<td>Session/visit duration</td>
<td>[ \begin{align*} \text{sessions per unique visitor} \end{align*} ]</td>
</tr>
<tr>
<td>Page views per session/depth of visit</td>
<td>[ \begin{align*} \text{sessions per unique visitor} \end{align*} ]</td>
</tr>
<tr>
<td>Visit frequency</td>
<td>[ \begin{align*} \text{sessions per unique visitor} \end{align*} ]</td>
</tr>
<tr>
<td>Visit recency</td>
<td>[ \begin{align*} \text{number of days elapsed since last visit} \end{align*} ]</td>
</tr>
</tbody>
</table>

Table 6.4: Overview of web usage metrics [Zumstein, 2012]
6.4.1 Model Preparation – Customer Satisfaction

In order to measure customer satisfaction without an active participation of customers, suitable SQMs need to be identified that are collectible in a passive, non-invasive way.

As most customer satisfaction evaluation techniques are based on surveys [McColl-Kennedy and Schneider, 2000], there is a lack of research on alternative ways to measure customer satisfaction in online service contexts.

For this reason and since there are scientific indications for their meaningful application in the customer satisfaction domain, we decide to proceed with measurement scales that were originally developed to assess service quality (e.g. [Benlian et al., 2011]).

Therefore, in a first step existing service quality measurement scales need to be identified, selected, validated and potentially adjusted to domain-specific aspects. This process results in service quality dimensions for the respective service domain (e.g. as presented in Table 6.1).

Once the service quality dimensions are defined, SQMs for each dimension are derived...
from the available data sources. Consequently, each dimension is represented by a set of SQMs. Based on the assumption that there exits an interrelation between service quality and customer satisfaction [McDougall and Levesque, 2000], the SQMs can be aggregated to a dimensional customer satisfaction score. These dimensional scores are then merged to a total score formula. Therefore, a weighting of the dimensions is advisable which can be determined based on expert interviews in the respective service domain.

6.4.2 Model Preparation – Web Usage Mining

In parallel to preparing the model on the customer satisfaction side, a foundation for extracting web usage metrics must be set up. For one, suitable metrics for characterizing the usage behavior of users and customers on B2B online platforms need to be identified in order to draw conclusions from usage data on customer satisfaction. In B2B contexts it is important to consider that customers and users are two distinct entities with a relation of 1:n. Therefore, the behavior of one customer is characterized by the usage behavior of its entire user base.

The metrics suitable to measure website usage are derived from existing literature (e.g. [Zumstein, 2012]). These metrics are mostly based on web server logs which are extracted, preprocessed and stored in a user transaction database, following the process presented in Section 6.3.3.

6.4.3 Model Application

In the model application phase, the customer satisfaction scores of customers are calculated by applying the score formula. Depending on their customer satisfaction score, customers are assigned to customer satisfaction classes that comprise customers showing a similar level of satisfaction. To guarantee meaningful results, the customer satisfaction scores need to be validated. Since it is often not possible to assess the real level of satisfaction directly from the customers for validation, we choose the usage data of customers and their end users as an alternative validation technique as usage behavior is an empirically proven consequence of and directly related to customer satisfaction [Bolton and Lemon, 1999].

In more detail, the web usage metrics of each user are calculated and merged on a customer level. These metrics are then linked to the respective customer satisfaction class of that customer. By aggregating the usage metrics within each of the classes, potential differences in behavior between classes can be identified. Since previous research found empirical evidence for a relationship between high levels of customer satisfaction and high service usage rates [Bolton and Lemon, 1999], we anticipate that the web usage metrics assigned to classes comprising satisfied customers indicate higher usage rates as compared to classes consisting of dissatisfied customers. In order to validate the customer satisfaction scores, classification techniques can be applied to the web usage metrics and their customer satisfaction class and the resulting accuracy can be used to estimate the meaningfulness of the score.

Ultimately, this leads to two main advantages: For one, customers can be assigned
to customer satisfaction classes based on their usage behavior. This knowledge can, for instance, be used by sales teams to proactively approach dissatisfied premium customers or very satisfied freemium customers to either prevent churn or sell more licenses respectively. For another, the insights on usage behavior related to customer satisfaction allow product managers to monitor how recent releases affected customer satisfaction and to, consequently, prioritize development activities accordingly.

6.5 Case Validation

In order to validate the previously presented model, it is instantiated in a real-world B2B online platform.

6.5.1 Model Preparation

Web Usage Metrics

All relevant information related to user behavior of a customer is stored in the platform’s usage transaction database. This includes user and customer information, license information as well as usage data (e.g. sessions IDs, session duration, page views etc.). In addition to that, there is a separate database containing complaint ticket data. The available information is aggregated to web usage metrics as presented in Table 6.4. Next, the metrics are normalized to guarantee the comparability of the usage behaviors on customer level.

Selection of Service Quality Measurement Scale

As existing service quality measurement scales are usually bound to a specific context, it is essential to select a scale that fits best to the domain-specific aspects of the case. The selected case comprises the following domain-specific aspects: 1) information systems context, 2) online service, 3) SaaS context, 4) B2B market, and 5) healthcare industry.

Consequently, we compare the domain-specific aspects of the case to the characteristics of the measurement scales presented in Table 6.2. The SaaS-QUAL measurement scale proves to be the best fit since both, SaaS and B2B, are covered in the service context and it was validated cross-industrially [Benlian et al., 2011].

Validation of SQ Scale

After selecting the SaaS-QUAL instrument, its applicability to the case must be validated. This validation is done via semi-structured expert interviews. We select the platform’s digital solution experts (DSE) as a suitable expert sample to validate the measurement scale for our case. They are the first contact partner for the platform’s customers and are technically trained by the development team. Each of the DSEs is responsible for a specific business region consisting of several countries. For the interviews, six DSEs are selected who are in total responsible for nearly 800 customers
in 18 countries.

As part of the interviews, the experts are asked to rate the influence of the SaaS-QUAL dimensions (see Table 6.1) on customer and user satisfaction. On a five-point Likert scale, the experts consider the influence of all six SaaS-QUAL dimensions on customer satisfaction between 3.5 (flexibility), 4.17 (reliability), 4.33 (responsiveness and features), 4.6 (rapport), and 4.67 (security) on average. Except for flexibility, the dimensions are all perceived to have a high or very high impact on the platform’s customer satisfaction. Even the average score of flexibility subtracted by the standard deviation expresses at least an intermediate impact.

**SQMs for Customer Satisfaction Measurement**

Based on the predefined service quality dimensions, SQMs need to be identified that can be used to build a score formula for measuring customer satisfaction.

**Rapport SQMs** The *rapport* dimension focuses on a service’s customer support and its quality. As a result, the number of complaint tickets play an important role in this dimension. To ensure comparability, the number of opened tickets must be normalized by the period a customer is registered. The following SQMs related to *rapport* were identified (CSAT_RAP):

- Tickets per month (CSAT_RAP_TPM)
- Open tickets (CSAT_RAP_OT)
- Closed delayed tickets (CSAT_RAP_CTD)
- Open delayed tickets (CSAT_RAP_OTD)

**Responsiveness SQMs** The *responsiveness* dimension measures the availability and performance of the online service. Two of the most important measures in this category are latency and availability. Latency is defined as the time that a service needs to return a response and availability is the ratio of requests that returned a successful response. As a result, the following SQMs for *responsiveness* were identified (CSAT_RES):

- Availability (CSAT_RES_AVA)
- Latency (CSAT_RES_LAT)

**Reliability SQMs** The *reliability* dimension aims at providing services in time, dependably and accurately. An important aspect of delivering the services of the platform’s applications is the installation and setup of the receiver at customer sites. The installation time per customer should not exceed 21 days and the receiver must work properly, otherwise the information shown to the customer may be inaccurate. Therefore, the identified SQMs for *reliability* are (CSAT_REL):
• Installation processes (CSAT_REL_INST)
• Receiver ticket complaints (CSAT_REL_TIC)

Flexibility SQMs  The freedom to change contractual and technical aspects of the relationship with a service provider is defined as the flexibility dimension. This includes requests for new functionalities summarized as feature request tickets. In addition to that, every time a customer wants to change the contractual relationship with the platform provider, a ticket is created. The resulting SQMs are (CSAT_FLE):

• Ratio closed feature requests (CSAT_FLE_FR)
• Resolve time for contract changes (CSAT_FLE_CON)

Features SQMs  The features dimension describes how well the functionality of a service meets the requirements of a customer. This can be measured for either existing features, or for missing or expected features. The resulting SQMs for the features dimensions are (CSAT_FEA):

• Feature request tickets (CSAT_FEA_REQ)
• Existing feature tickets (CSAT_FEA_TIC)

Security SQMs  Tickets concerning security and privacy of the platform are assigned to the security dimension. Due to the limited logging of information, we were only able to identify one SQM within the available data sets for the security dimension (CSAT_SEC): Security tickets (CSAT_SEC_TIC)

Customer Satisfaction Score Formula

The identified SQMs for each dimension can now be merged into a overarching score formula. This can be done on two levels of granularity: For one, the normalized SQMs of a single dimension can be combined to calculate a weighted average for that dimension individually, resulting in dimension-specific satisfaction scores. In a first step and due to the lack of empirical evidence, all SQMs are weighted equally and the factors are set to 1.

For another, the overall customer satisfaction score can be calculated by combining the previously calculated satisfaction scores of the individual dimensions. Analogously to the previous calculations, a weight can be assigned to each of the dimensions in order to reflect the influence of the dimensions on customer satisfaction. The normalized average ratings derived from the expert interviews can be used to determine a first set of weights.

The dimensional weights, therefore, are: 1) Rapport: 0.18, 2) responsiveness: 0.169, 3) reliability: 0.163, 4) flexibility: 0.137, 5) features: 0.169, and 6) security: 0.182 As a result, the final customer satisfaction score formula is: \( CSAT_{score}(c = customer) = 0.18 * CSAT\_RAP(c) + 0.169 * CSAT\_RES(c) + 0.163 * CSAT\_REL(c) + 0.137 * \)
CSAT\_FLE(c)+0.169*CSAT\_FEA(c)+0.182*CSAT\_SEC(c). The calculation process is illustrated in Figure 6-5.

### 6.5.2 Model Application

#### Calculation of CSAT scores

The previously assembled formula is now used to calculate the \textit{customer satisfaction score} for each of the platform’s customers. Initially, this is done for the dimensions individually before merging the results to an overall satisfaction score.

Figure 6-6 provides an overview of the mean and median scores of the respective SQMs after normalization. The calculation of SQMs for individual customers is highly dependent on the available data. The \textit{flexibility} dimension, for instance, relies on either closed feature request tickets (CSAT\_FLE\_FR) or contract change tickets (CSAT\_FLE\_CON). In many cases, the customer did not open a ticket in either of these categories. Therefore, the respective SQMs could not be calculated which is the reason for the varying number of customers.

The calculated SQMs can now be merged on a dimensional level to calculate the \textit{dimensional customer satisfaction scores}. During the calculation of the scores for each customer, respective SQMs are only taken into consideration if the SQM is available for that customers. To ensure comparability, the score is normalized according to the number of available SQMs. The results of the \textit{dimensional customer satisfaction scores}...
When comparing the distributions of the dimensional scores, it stands out that CSAT_RAP, CSAT_RES, and CSAT_REL all consist of more than 1000 customers and their standard deviation is below 0.2. The other dimensions have higher standard deviations and customer satisfaction levels differ more. An explanation for the higher variation is most likely the lower number of customers as the calculation base. Except for CSAT_FLE, the dimensional distributions are concentrated mostly above the 0.5 level, whereas CSAT_FLE values are concentrated on the edges. This implies that customers are either satisfied or dissatisfied with **flexibility**.

After calculating the *dimensional customer satisfaction scores* for each customer, they are merged into overarching *customer satisfaction scores* by applying the previously presented formula. The weights of the formula are normalized according to the availability of dimensional scores. On average, the customer satisfaction score consists of 2.58 dimensions with a median of three dimensions. The distribution of customers across customer satisfaction scores is presented in Figure 6-8. It is a left-skewed curve, which is typical for customer satisfaction distributions [Peterson and Wilson, 1992]. Many customers tend to have high customer satisfaction scores, while only a few are dissatisfied.
Customer Satisfaction Classes

In the next step of the model, customers are assigned to customer satisfaction classes based on their customer satisfaction score. We explore three different techniques for classifying customers based on their score: 1) 5-point Likert scales to assign customers to one of five classes from very dissatisfied to very satisfied; 2) Perc5 to group customers by the percentiles 20, 40, 60, and 80; and 3) Bin3 classification to distinguish between satisfied, neither satisfied nor dissatisfied, and dissatisfied customers. For our case, we accept the Bin3 classification as the best solution. While the sample size of the dissatisfied customers is smaller than the ones of the other groups, the interval sizes are on average higher compared to the other approaches. Class 1 comprises dissatisfied customers \((n = 4)\) with a customer satisfaction score interval of 0 to 0.4, while neither satisfied nor dissatisfied customers \((n = 175)\) are represented by class 2 with a score interval of 0.4 to 0.6 and satisfied customers \((n = 1656)\) with a score interval of 0.6 to 1.0 are assigned to class 3.

Usage Behavior Metrics

In order to trace the satisfaction of a customer back to its behavior, we extract the usage metrics of 1400 customers. For this, we stick to the metrics already established in existing literature (see Table 6.4). Table 6.5 presents the mean, median and standard deviation of the web usage metrics calculated for the customers of the platform and its applications. Due to very high variations in the data, the standard deviation of some metrics is extremely large. The results indicate that the usage behavior of customers is very diverse, for instance
the standard deviation of the page views metric is nearly twice as high as its mean. The usage activity among customers on average is rather low with 0.6 monthly active users and 1.7 sessions per user and month. On average users spend nearly 5 minutes in one session and the average recency of about 78 days implies that many customers do not use the platform and its applications on a regular basis.

Usage Behavior of Satisfaction Classes

We examine the characteristics of the individual metrics within the respective classes in order to find out whether customers of different classes also show a difference in usage behavior. Due to the limited customer base of unsatisfied customers ($n = 4$), we only focus on classes 2 and 3 (neither satisfied nor unsatisfied and satisfied) in the upcoming analysis. Figure 6-9 visualizes the boxplots of four web usage metrics in classes 2 ($NEI$) and 3 ($SAT$).

The usage metrics related to page views (page views, unique page views, and page view duration) show a significant difference between the two classes and imply a more intense usage behavior of satisfied customers. Whereas the mean page views of class $SAT$ are about 248 (and significantly below the global mean), the average of class $NEI$ is about 445. The median of class $SAT$ (174) is more than twice as large as the median of class $NEI$ (see Figure 6-9a).

Similar results can be observed for metrics related to the number of users of customers.
Table 6.5: Calculated web usage metrics of 1400 customers

For instance, the mean and median values for the monthly active users per customer are significantly higher in class SAT as compared to class NEI (see Figure 6-9b). While some metrics related to the sessions of a customer do not show any significant differences between the two classes (e.g., average page views per session or average session duration), others indicate a clear difference between classes SAT and NEI. The mean and median number of overall sessions is twice as high in class SAT as in class NEI (see Figure 6-9c). Also the visit frequency of satisfied customers appears to be higher compared to neither satisfied nor dissatisfied customers (see Figure 6-9d).

Customer Satisfaction Classification

The last step of the model is to close the feedback loop between customer satisfaction and usage behavior. Specifically, we aim at validating the meaningfulness of the customer satisfaction scores by classifying the usage behavior to the customer satisfaction classes. This approach is especially useful for cases without access to direct customer
feedback and relies on the scientific evidence of a relation between high service usage and high customer satisfaction [Bolton and Lemon, 1999].

To achieve this classification, the previously extracted web usage metrics on customer level are now labeled with the respective customer satisfaction class. Next, several decision trees are created with different parameter values to classify customers into the Bin3 customer satisfaction classes based on their usage behavior. For the training of the decision tree and the following prediction, the customers in the classification matrix are split into a training and a test dataset. In total 72 decision trees were trained with different parameters, including test size (20% and 30%), minimum samples per leaf (1, 2, and 3), maximum depth of the tree (5 to 10), and the selected web usage metrics (all vs. metrics with significant differences between classes). The best-performing decision tree reaches a prediction accuracy of 91.2% with a test size of 30%, a maximum tree depth of 5, and the web usage metrics page views, unique page views, page view duration, number of users, monthly active users per user, weekly active users per user, daily active users per user, sessions, visit frequency, and minimum recency. The three most important features for the customer satisfaction classification are sessions, weekly active users per user, and unique page views.

Based on the significant difference in usage behavior across classes presented in the previous section (see Figure 6-9) as well as the accuracy score of 91.2% we have evidence to believe that the calculation of customer satisfaction scores, which is used to divide customers into the respective classes, provides reasonable results to measure customer satisfaction.

### 6.6 Related Work

Service usage is considered a consequence of customer satisfaction [Bolton and Lemon, 1999]. Specifically, Bolton et al. empirically prove a quantitative relationship between high satisfaction levels and high service usage. Customer satisfaction consequences, such as repurchases and revenue, can in turn be used to measure customer satisfaction [Grigoroudis and Siskos, 2009].

Moreover, Neelima and Rodda [Neelima and Rodda, 2015] claim that conclusions from web usage mining can lead to improved customer satisfaction. Spiliopoulou and Pohle [Spiliopoulou and Pohle, 2001] propose a model for measuring a website's (B2C) success through web usage mining. Although their research implies that a connection between customer satisfaction and success may exist, a concrete dependency is not outlined [Spiliopoulou and Pohle, 2001].

Another rationale for the interrelation between customer satisfaction and web usage mining is that by applying web usage mining techniques, a website can be improved in various ways [Zumstein, 2012]. These improvements, in turn, are likely to positively influence customer satisfaction. The focus of these studies is typically on web usage mining for B2C websites, resulting in a lack of research on web usage mining for B2B service providers. Due to the complex relationships between customers and end-users in B2B, we argue that it is
equally important to investigate how principles of web usage mining can be leveraged to improve customer satisfaction in B2B contexts, which is also the main goal of our study.

6.7 Threats to Validity

One threat to the external validity of our study is the limited number of applied measurement scales as the validation in the case company only proves the applicability of the SaaS-QUAL measurement scale. Therefore, the successful application of other service quality scales cannot be guaranteed.

In addition to that, the SQMs used to calculate the customer satisfaction score are highly dependent on the available data collected by the platform provider. Other types of collected data may result in different SQMs, different score formulas and, consequently, different customer satisfaction scores.

Moreover, the weighting of the customer satisfaction dimensions and the validation of the SaaS-QUAL scale are based on expert interviews. Although these experts are equipped with a significant amount of business experience and are the direct point of contact for the customers, they can only partially represent customers’ perceptions and potential bias cannot be ruled out. Additionally, the sample size of six experts is rather small. Therefore, the extracted knowledge, such as the dimensional weights, cannot be generalized. This constitutes a threat to the internal validity of our study.

6.8 Conclusion

Improving customer satisfaction in B2B online service domains is crucial for service providers. As receiving high-quality information about customer satisfaction usually involves extensive effort, our model aims at measuring customer satisfaction in an automated and quantitative way that does not require active customer participation.

The presented model is derived from scientific literature using a deductive research approach and its applicability is proven by instantiating it in a real-world B2B online platform. The results indicate a meaningful assessment of customer satisfaction based on quantitative SQMs. This assumption is supported by the fact that we were able to identify a significant difference in usage behavior across customers depending on their satisfaction score.

While previous research indicates a connection between customer satisfaction and usage behavior, to our knowledge so far no attempt has been made to measure customer satisfaction in a fully automated and quantitative way based on usage data. Therefore, the main advantage of our approach is that it does not require any effort on the customer side, the outcome is easy to interpret due to concrete SQMs and numeric scores, and it can be executed continuously, even allowing real-time monitoring of customer satisfaction.

One limitation of our study is the application of service quality measurement scales in the context of customer satisfaction. Although there are indications in existing
literature that the two domains are interrelated, a direct transferability is yet to be proved. A second limitation is, despite the fact that usage behavior has been identified as a consequence of customer satisfaction [Bolton and Lemon, 1999], we cannot prove with certainty that customers who use the platform and its applications more intensively are really more satisfied.

Future research could be dedicated to further validate the model and to investigating the relationship between usage behavior and customer satisfaction more closely in the context of customer satisfaction measurement.
Chapter 7

Breaking the Vicious Circle: A Case Study on Why AI for Software Analytics and Business Intelligence Does not Take off in Practice


Abstract

In recent years, the application of artificial intelligence (AI) has become an integral part of a wide range of areas, including software engineering. By analyzing various data sources generated in software engineering, it can provide valuable insights into customer behavior, product performance, bugs and errors, and many more. In practice, however, AI for software analytics and business intelligence often remains at a prototypical stage, and the results are rarely used to make decisions based on data. To understand the underlying causes of this phenomenon, we conduct an explanatory case study consisting of an interview study and a survey on the challenges of realizing and utilizing artificial intelligence in the context of software-intensive businesses. As a result, we identify a vicious circle that prevents practitioners from moving from prototypical AI-based analytics to continuous and productively usable software analytics and business intelligence solutions. In order to break the vicious circle in a targeted manner, we identify a set of solutions based on existing literature as well as the previously conducted interviews and survey. Finally, these solutions are validated by a focus group of experts.
7.1 Introduction

Artificial intelligence (AI) has been known as a powerful tool to extract valuable information from data for quite some time. Consequently, this led to a growing interest of software-intensive companies in performing software analytics and business intelligence based on Artificial Intelligence (AI4SABI) in order to acquire meaningful and relevant information about their products and processes. Many advances have already been made in the analysis of software engineering data and data-driven product development [Olsson and Bosch, 2019], [Buse and Zimmermann, 2012]. For instance, customer behavior and product usage are being analyzed to provide insights for developers, software architects, product managers and even business-oriented roles such as sales [Figalist et al., 2019a], [Figalist et al., 2019b].

However, if no measures or actions are initiated based on its results, the analysis of customer behavior remains insufficient [Butz Jr. and Goodstein, 1996]. Prototypically implementing an analysis is simple, but in order to truly base decisions on it, an analysis must be delivered continuously and automatically. After providing AI4SABI solutions in [Figalist et al., 2019a], [Figalist et al., 2019b], and [Figalist et al., 2021], we observed on several occasions that analyses often get stuck at this point and are not pursued further.

To address this situation, our study focuses on identifying challenges and appropriate solutions for the application of data analytics and in particular AI in the context of software analytics and business intelligence. In order to identify the challenges and the underlying causes of our observation, we conduct an explanatory case study consisting of an interview study and a survey. Thereby we investigate two different perspectives, namely that of the (potential) users (e.g. product managers, software architects, sales) and that of providers (e.g. data scientists, data engineers, operations engineers, software architects) of AI4SABI solutions. This work builds on an earlier publication [Figalist et al., 2020] that only covers the challenges. In this paper we now add to it by additionally reviewing existing literature as well as the conducted interviews and survey for solutions that address the presented challenges. We categorize the identified solutions and conduct expert interviews for a validation of the same.

The contribution of this paper is three-fold. First, we identify the challenges for applying AI4SABI from two different perspectives, specifically the users and the providers of AI4SABI. This contributes to an improved mutual understanding. Second, we introduce the concept of a vicious circle consisting of key drivers why AI4SABI is rarely realized in a productized way. This circle supports involved stakeholders in comprehending why their AI4SABI solutions might get stuck in a prototypical stage. Third, we present a set of solutions mapped to the key drivers in the vicious circle to support practitioners in taking a targeted approach at breaking the circle.

The remainder of the paper is structured as follows: First, the background of this study is provided in Section 7.2. The research method and study design is described in Section 7.3, while the findings of the interview study and the survey are presented in Sections 7.4 and 7.5. In section 7.6 the derived challenges forming a vicious circle are outlined before investigating and presenting potential solutions in Section 7.7. The
threats to validity are discussed in Section 7.8 followed by the conclusion in Section 7.9.

### 7.2 Background

The following subsections give an overview of the existing literature in the fields of software analytics and data-driven software engineering as well as the challenges regarding data analytics and AI. Both topics are already quite well researched as individual areas. However, in our study, we feel it is important to put both areas into a common context, since the products analyzed by software analytics or business intelligence are often not related to AI or data analysis, which generally impedes a successful implementation of AI4SABI.

#### 7.2.1 Software analytics and Business Intelligence

The term software analytics (SA) depicts the application of data analysis to software data in order to generate insights for various stakeholders, from software engineers to managers, that ultimately support their decision making [Buse and Zimmermann, 2012], [Menzies and Zimmermann, 2013]. Related to this, the field of business intelligence (BI), sometimes also referred to as business analytics, utilizes data mining techniques on operational data to derive information for managerial decision making [Negash and Gray, 2008].

Analyzing customer behavior can be a valuable asset in making strategic decisions on product improvement and evolution [Marciuska et al., 2014], [Olsson and Bosch, 2013]. One of the most critical challenges is choosing the right metrics to serve a specific purpose or achieve a specific goal. Therefore, each metric must be carefully evaluated in terms of its suitability and explanatory power [Kaner et al., 2004]. At the same time, it is considered crucial, but also difficult, to ask the right questions and specify information needs [Olsson and Bosch, 2015], [Figalist et al., 2019a]. In order to really create impact by acting on the results, it is decisive to select and prioritize the right questions [Kim et al., 2016]. Different roles have very different information needs, and while the information needs of technical stakeholders are well-researched, the needs of managers are less transparent [Buse and Zimmermann, 2012]. Moreover, the visualization of data and results in an interpretable way has proven to be quite complex, but nevertheless crucial to generate results in an understandable and explainable way. This is essential to establish a certain level of trust and comprehensibility required to act on the basis of the provided information [Buse and Zimmermann, 2012], [Menzies and Zimmermann, 2018], [Dam et al., 2018], [Kim et al., 2016]. Furthermore, there is still a lot of skepticism among stakeholders who do not trust the data and the analyses’ outcomes, and feel like going data-driven is risky and time-consuming [Olsson and Bosch, 2019].

From data generation to data collection, data storage, data analysis, visualization, and ultimately to value creation and smart decision making, it is a long and complex process [Saggi and Jain, 2018]. In software-intensive businesses, data is generated from
various sources. It can be machine-generated (e.g., network data), human-generated (e.g., log or usage data) or business-generated (e.g., transactional data). The distribution of available data sources is often accompanied by a lack of sharing data among individual teams [Fabijan et al., 2016], [Cito, 2016]. In addition to that, all data sources have to be understood and quality issues must be resolved. Following this, a variety of preprocessing techniques are required (e.g., cleaning, integration, transformation) before an analysis can be performed that generates meaningful insights [Saggi and Jain, 2018], [Hu et al., 2014]. Throughout this entire process it can be challenging to deal with complex types and structures of data from multiple sources, to filter, process and store it in a usable format, and to select appropriate algorithms to analyze it [Saggi and Jain, 2018].

7.2.2 Artificial Intelligence – Definition and Challenges

Artificial intelligence can be defined as "the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience" [Copeland, 2020]. Machine learning (ML) is "a type of artificial intelligence in which computers use huge amounts of data to learn how to do tasks rather than being programmed to do them" [Oxford, 2021]. ML can be applied to various problems, such as text classification, natural language or speech processing, or computer vision [Mohri et al., 2018], [Jordan and Mitchell, 2015]. Standard learning tasks of ML include classification, regression, ranking, and clustering [Mohri et al., 2018].

With its increase in popularity, artificial intelligence soon became an integral part of SA and BI solutions [Chen et al., 2012], [Dam et al., 2018]. In the context of our study, we define AI4SABI as the application of artificial intelligence methods to software or operational data in order to address problems or generate insights in the area of software analytics or business intelligence. In recent work, we have implemented a variety of use cases for AI4SABI, such as recommendation systems for API refactoring based on genetic algorithms or customer churn prediction models based on operational usage data using classification techniques [Figalist et al., 2021]. However, working with large amounts of data and applying complex algorithms to it often entails a number of challenges. Three types of AI-related challenges have been identified in the existing literature: development-, production- and organization-related challenges [Arpteg et al., 2018]. For instance, a lack of transparency makes it difficult to understand machine learning models, debug code and estimate effort. Careful monitoring of frequent updates is required, especially with regard to changes or additions in data sources. Furthermore, building and using machine learning systems usually depends on a variety of components (e.g. infrastructure, data pipelines, visualizations). As a result, it often requires a collaboration of several teams and roles within teams that do not always share the same ideas and priorities. Therefore, it can be very complicated to align the teams and overcome cultural differences [Arpteg et al., 2018].
Similarly, a large-scale study of data scientists reveals that the key problems they face are mostly either data (quality, availability and preparation), analysis (scaling and evaluating different models) or humans (communicating results and convincing teams of their value) [Kim et al., 2017]. Moreover, the code that analyzes data represents only a very small fraction (5%) of the overall system. A significant amount of effort is also put into configuration, data collection, feature extraction, data verification, infrastructure, monitoring and other activities [Sculley et al., 2015].

7.2.3 Prototypes in Data Science Projects

Oftentimes, data science projects start out as prototypical implementations to try out new ideas and get some first results fast [Provost and Fawcett, 2013], [Sculley et al., 2015]. This also applies to data science projects in software analytics [Misirli et al., 2013] and business intelligence [Thierauf, 2001]. A major downside of this is that the results of small-scale prototypes often do not reflect the results in reality [Sculley et al., 2015]. After implementing several use cases of AI4SABI in [Figalisto et al., 2021], we observed that, on the one side, prototypes are a powerful tool to explain and convey ideas to stakeholders as an analysis becomes more tangible if there are some first results to look at. On the other side, the majority of analyses only provide the desired value if executed in an automated and continuous way (e.g. customer churn prediction models). However, the transformation of a prototypical analysis into an automated and deployed analysis is difficult and time-consuming [Misirli et al., 2013], [Sculley et al., 2015]. This results in "prototypes or early models that lack the final deployment step, and hence, any real-world impacts" [Misirli et al., 2013].

7.3 Research Method & Study Design

In order to investigate the application of AI4SABI in a real-life contexts, case study research was selected as the research method for our study. We conducted an explanatory case study that was designed by following the guidelines presented by Runeson and Höst [Runeson and Höst, 2009]. The case study consisted of a qualitative interview study and a quantitative survey. The combination of qualitative and quantitative research methods has proven to be a successful means of obtaining useful information that could not be obtained from individual methods alone [Seaman, 1999]. For this reason, we applied a mixed-method approach in our study following a concurrent triangulation strategy [Easterbrook et al., 2008]. The objective of the study is defined as 1) the identification of challenges and key drivers why AI4SABI is rarely productized (automated and deployed) and utilized for decision-making in practice and 2) the identification of potential solutions to address these challenges. As a result, the studied case is the application of AI4SABI within real-life product teams. We aim at addressing the following research questions:

1) RQ1: What are the key drivers for analyses not being productized after the prototypical stage?
2) RQ2: How can the identified challenges and key drivers be addressed?

After summarizing the results of the interview study and the survey, each of the research questions is addressed in a dedicated section (Sections 7.6 and 7.7). The case company is a large, industrial company with around 293,000 employees and multiple subsidiary companies. The case company and its subsidiaries offer a diverse product portfolio in a variety of different domains, such as industry, infrastructure, mobility, healthcare, and energy. It is an international company with product teams that are spread all over the world (mostly in Europe, USA, India, and China). In order to identify the challenges, we conducted both an interview and a quantitative survey as part of our case study. We then compared the results in order to confirm and cross-validate our findings.

In a next step, we reviewed existing literature and the previously conducted interviews and survey for potential solutions. These solutions were then categorized, linked to the challenges and validated by a focus group of experts. The overall research process is illustrated in Figure 7-1.

![Figure 7-1: Research process](image)

7.3.1 Interview Study

The objective of the interview study is equal to the objective of the overarching case study. Semi-structured interviews were selected as a direct form of data collection. Specifically, we interviewed twelve stakeholders who are either users or providers of AI4SABI. Each participant either belonged to one of two products (*Product A* or *Product B*) or is a consultant to a number of product teams.
Product Descriptions

Product A is a platform provider operating in the healthcare domain. The platform hosts several platform-internal and external medical applications for its ∼2700 customers. Subproducts A₁₋₃ in our interviews represent internal applications of the platform. The platform and its internal applications are developed and operated by six modularized teams with sizes varying between 5 and 25 team members. Each team is either responsible for a specific platform component or a specific application that is hosted on the platform. The teams use Azure DevOps¹ and its services for the development and operations of their product. Moreover, they have a dedicated operations team that collects performance, usage, and sales data of the platform and its applications in order to create dashboards for different stakeholders and make it available to all other teams for further analysis. They use Microsoft Power BI² for the creation of dashboards and visualization of their analyses. They use both, business intelligence and software analytics, to generate insights for their stakeholders. The majority of their analyses are not AI-based. For example, as part of their business intelligence dashboards for their sales team, they provide statistics of the usage rates of customers clustered by the time to their subscription’s expiration date. They use software analytics to generate information for their technical stakeholders, e.g. developers and architects. As an example, they analyze and visualize the navigation flows of customers to support the development teams in understanding whether their component or application is used as intended. AI-based examples that they have recently started working on are customer churn prediction models based on usage data and customer satisfaction classification based on service quality and web usage metrics. They have successfully productized their non-AI-based analyses and are now in the process of doing the same for their ML models. In the future, they aim at expanding their analyses and fully productizing their models.

Product B is a platform provider operating in the industrial domain. The platform serves ∼800 customers and hosts multiple company-internal and external applications for industrial device management and analysis. Similar to Product A, the platform and its applications are being developed by modularized and distributed teams. In total, around 330 individuals are involved in the development of the platform with similar team sizes as Product A. The teams are following DevOps principles for the development and operation of their platform and applications. The platform provider setup a dedicated team for logging, processing and storing all requests users make to the platform and its applications. Their goal is to enable the other teams to extract continuous feedback based on the usage of their platform component or application. The platform’s sales team works with a team of data scientists that creates BI dashboards displaying statistics of the customers’ usage and sales data. In addition to that, a customer churn prediction model was trained based on the customers’ usage data. However, the latest predictions of the model are not yet integrated in the dashboard. In the area of SA, the platform’s security team worked with a company-internal research team who applied pattern detection to the usage data (specifically

¹https://azure.microsoft.com/services/devops/
the requests authentication headers) in order to detect wrongfully assigned permissions. To use the analyses on a continuous basis, the stakeholders would like to automate and deploy the analyses in the near future.

### Interviewees

The interviewees can be categorized into two groups: users or providers of AI4SABI. In order to identify the interviewees for our study, we sent out a description of the study and an invitation to participate to contact persons of various platforms and applications within the case company who shared it among their teams. We received twelve responses from ten different teams distributed across two platform providers. The interviewees were selected due to their interest in either using or providing AI4SABI solutions. All interviewees from the provider group had already experience in applying AI in the context of SA or BI. In the user group, not all participants had hands on experience in using AI4SABI. However, this was not an exclusion criteria for us since the reasons for this were also relevant for our case study. From the user perspective we interviewed two product managers, three product owners, one demand manager, and one software architect while the provider perspective was investigated by interviewing one product owner (for operations), one operations engineer, one data scientist, and two software architects. An overview of the interviewees is summarized in Table 7.1.

Before the interviews were conducted, two interview guides were prepared, one for each group (users and providers). In addition, each interview guide was slightly adapted to the respective role of the interviewee. The interview guides for the users of AI4SABI covered the following topics: current process of decision making in their job; current status of using data as feedback and how data and information is collected and shared within and across teams; and experiences with and views on data-driven feedback and decision making. The interview guides for the providers included: the process of building and providing AI4SABI solutions (including data collection and processing, implementation, automation and deployment of analysis, preparation of results); the organizational and human-related challenges they face; and technical difficulties they encounter.

<table>
<thead>
<tr>
<th>Users of AI4SABI</th>
<th>Product A</th>
<th>Product B</th>
<th>Consultant (multiple products)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product manager</td>
<td>Product owner (Sub-product A₁)</td>
<td>Product owner</td>
<td>Software architect (i.a. working for Product B)</td>
</tr>
<tr>
<td>Product manager</td>
<td>Product owner (Sub-product A₂)</td>
<td>Demand manager</td>
<td>[i.a. working for Product B]</td>
</tr>
<tr>
<td>Product owner (Sub-product A₃)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Providers of AI4SABI</th>
<th>Product owner (for operations)</th>
<th>Software architect (i.a. working for Product B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations engineer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data scientist</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.1: Interviewees grouped by products and function
Interview Process

At the beginning of each interview, the interviewee was asked for permission to record the interview. The interviews lasted between 35 minutes and one hour, and were transcribed afterwards. We applied a thematic analysis approach [Maguire and Delahunt, 2017] for analyzing and coding the interview transcripts. The objective was to identify key challenges in working in a data-driven way that prevent a successful application of AI4SABI. The high-level themes used for thematic coding were: 1) current decision-making processes; 2) current state of data collection; 3) current state of data analysis; 4) perceived usefulness of AI4SABI; 5) mindset towards data-driven decision-making; 6) people-related challenges (e.g. skepticism, interpretability); 7) organizational challenges (e.g. priority, resources); and 8) data-related challenges (e.g. data model, data quality). The coding itself was performed by the first author. The results were shared and discussed with the other authors to avoid any misinterpretations. Finally, the findings of each interview were summarized and sent to each interviewee for further validation.

We achieve triangulation by: 1) interviewing multiple roles with opposite points of view (users and providers of AI4SABI); 2) conducting interviews within multiple independent product teams; and 3) enriching the results by an additional quantitative data source (survey). In addition to that, the transcript and summaries were also shared and discussed among the researchers to get several interpretations and perspectives in order to avoid misunderstandings and to make sure to cover all important aspects.

7.3.2 Survey

For the execution of the survey we followed the guidelines of Kitchenham et al. [Kitchenham and Pfleeger, 2003] and Ciolkowski et al. [Ciolkowski et al., 2003]. In order to achieve the overall research goal, the survey focused on the current state of applying data-driven approaches in real-world projects, as well as on the experiences and challenges in providing and using customized AI4SABI solutions. The target population of the survey were stakeholders in the field of software engineering who are either (potential) users or providers of AI4SABI. To conduct the survey, we created an online questionnaire with a total of 29 questions that were divided into three parts: 1) personal background; 2) current status of projects with regard to data-driven approaches and analytics tools; and 3) experiences and challenges in implementing, automating, deploying, and leveraging AI4SABI. The first part consisted of three multiple choice questions regarding the participants’ role, their experience, and the tasks they perform as part of their job. The second part comprised 8 questions in total, two of them multiple choice (e.g. "Do you or others in your (main) project’s team use any analytics tools to gain insights from usage and/or runtime data?") and three single choice (e.g. "On a five point scale, how satisfied are you or the ones using it with the preparation and presentation of the results of these analytics tools?"). The third part consisted of 18
questions in total. One multiple choice question: "Do you or others in your (main) project’s team manually analyze data?". The answers to this question were divided into who in the team is manually analyzing data (the respondent or someone else in the team) and, if applicable, into the type of analytics that is being applied, e.g. basic analytics, such as simple statistics, or advanced analytics techniques, such as machine learning. In addition to that, the respondents were asked to rate certain challenges or the difficulty of certain tasks on a 5-point Likert scale (15 single choice questions, e.g. "On a five point scale, based on your experience, how difficult is the transformation of a prototypical analysis into an automated, continuous analysis?"). Finally, we included two open questions to gain insights in challenges or difficulties that we might have missed (e.g. "If you consider it hard or extremely hard, what do you consider particular difficult?").

In the third part, the results of the interview study were partially used as a basis for the survey questions, especially for rating the adverse factors on the application of data analytics in the respondent’s projects that were previously identified in the interviews and literature. The majority of the questions were based on 5-point Likert scales. As the backgrounds of the participants are very diverse, there was always the option to abstain if no knowledge in this area was available. We decided to send out the same questionnaire to both groups since a clear separation of the two is often not feasible. In fact, before sending out the questionnaire, we collected feedback on the questions within a department of the case company and learned that people often-times take on multiple roles. Therefore, they could not clearly be assigned to either the users or providers of AI4SABI since that depends on the individual role.

The link to the survey was sent via email to 1596 individual stakeholders of several divisions across our case company. We received a total of 230 responses which equals a response rate of 14%. The survey results were investigated from two angles. First, we described the distribution of answers for all single-choice and multiple-choice questions. The majority of questions were based on 5-point Likert scales (e.g. not at all, slightly, moderately, very, or extremely difficult / influential) which made them easy to compare. Second, we examined the responses in the free text fields and extracted and categorized relevant statements similar to the coding done in the interview study. For conducting the survey, we had to use the company-internal survey tool. Unfortunately, this tool does not allow an evaluation of results across questions but only provides the answers for each question independently. For this reason, we were not able to analyze the dependencies between questions and factors.

7.3.3 Derivation of Challenges and Key Drivers

In order to identify the challenges and key drivers why AI4SABI is rarely productized and utilized for decision-making in practice, we investigated and compared the results of the interview study and the survey. In a first step, the high-level themes identified during coding of the interviews were examined in more detail. All statements relevant for addressing our research questions were extracted for each of the themes. In a next step, we grouped the statements within each theme into further sub-themes (e.g. from data-related challenges to data-quality-related challenges). Most of
the challenges were identified in the themes people-related challenges, organizational challenges, and data-related challenges. However, other themes sometimes provided further explanations for certain situations and problems (e.g., in perceived usefulness of AI4SABI or mindset towards data-driven decision making). In these cases, the relations were also documented.

In a second step, we extracted and categorized the findings of the survey. For example, we examined the ratings of the twelve adverse factors and calculated the mean and median values of each response category across all factors. The mean values for the response categories very or extremely influential were 29.8% and 8.6%, while their median values were 28% and 9.5% respectively. Based on this, we set the thresholds for relevant factors to 30% and 10% for the categories very and extremely influential. Factors that exceeded one or both of the thresholds were highlighted as especially influential. In addition to that, the coded responses of the free text fields were grouped into subcategories similar to the sub-themes of the interview study.

Finally, we compared and mapped the results of the interview study to the results of the survey in order to cross-validate our findings. During the analysis, we observed that there are certain key challenges mentioned in the interviews and the survey that act as key drivers influencing each other. Based on this observation, we were able to identify a vicious circle that impedes a successful application of AI4SABI.

7.3.4 Identification and Categorization of Solutions

For the identification of potential solutions we used three different sources of information: existing literature, the interviews conducted as part of the interview study, and the survey responses.

In a first step, we queried common scientific libraries (IEEexplore, ACM Digital Library, ScienceDirect, Springer Link) using search terms related to the previously identified challenges. This process was conducted as a systematic literature review, but as a literature search in order to identify a list of key publications. Out of these publications, we extracted concrete actions that authors describe to improve situations related to the presented challenges as potential solutions and assigned each solution to the corresponding challenge.

Even though the interviews were primarily designed for identifying the challenges faced by practitioners, it also invited the interviewees to share their experience and expertise with regards to the actions and solutions they had found useful to address these challenges. Therefore, we examined the interview transcripts from an additional angle and extracted the proposed actions and solutions before also linking them to the respective challenges.

Similarly to the interview study, several survey respondents used the free text fields to share their experiences and solutions to specific problems. These solutions were also extracted and assigned to the challenges.

In a next step, we grouped the identified solutions of each challenge by their similarity and assigned related solutions to overarching solution categories.
7.3.5 Expert Interviews

In order to validate the solutions we identified, we interviewed six experts who have been working on different topics in the area of AI4SABI for multiple years. Four of them have already participated in the interviews of the previously conducted interview study. An overview of the interviewees and their qualification is provided in Table 7.2.

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Product</th>
<th>Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data scientist</td>
<td>Product A</td>
<td>Analyzes data &amp; communicates with stakeholders</td>
</tr>
<tr>
<td>Operations engineer</td>
<td>Product A</td>
<td>Provides infrastructure for AI4SABI</td>
</tr>
<tr>
<td>Operations manager</td>
<td>Product A</td>
<td>Builds dashboards for customer insights &amp; communicates with stakeholders</td>
</tr>
<tr>
<td>Product owner for operations</td>
<td>Product A</td>
<td>Intermediate between data scientists and stakeholders</td>
</tr>
<tr>
<td>Software architect [1]</td>
<td>Product B</td>
<td>Builds AI4SABI solutions [infrastructure analysis]</td>
</tr>
</tbody>
</table>

Table 7.2: Overview of expert interviewees

In the beginning of each interview the interviewees were given a short introduction into the background of our study and the challenges we had identified. Next, we presented the categorized solutions and explained how they relate to the respective challenges. Based on this, the experts gave their assessment of the solutions’ usefulness. After discussing the solutions in detail, the experts were also asked to rate the usefulness of the solution categories on a scale from 1 (not useful at all) to 5 (very useful). Each interview lasted between 30 minutes and one hour.

7.4 Interview Study Results

In order to identify the underlying causes of why AI-based analyses for SA and BI often get stuck after a prototypical implementation, we conducted twelve interviews with stakeholders involved in the software engineering process. The following subsections outline the perspectives of two groups (users and providers) and the challenges they face when trying to establish data-driven ways of working.

7.4.1 Group I - Using AI4SABI Solutions

The first group of interviewees consists of one software architect, one demand manager, two product managers, and three product owners. All of them are working in different teams of the two products. In addition to Product B, the software architect also works as a consultant for additional projects.

Recognizing Value

All of the interviewees are very open-minded towards AI4SABI and can see the value and benefits in developing software in a data-driven manner. For instance, the product manager of Product A explains that "sometimes what the stakeholders [within
the product teams] are asking for is not what the customer really wants". Therefore, data-driven feedback of customers would be a valuable asset for her to make decision on product or feature evolution. Furthermore, the product owner and demand manager of Product B both emphasize the difficulty of receiving early feedback from customers and, therefore, see the value in additionally collecting and analyzing data-driven feedback. A very similar statement is also given by the product owner of Subproduct A₂ and the product manager of Subproduct A₂ who would like to have more information on how customers are using the product’s features.

Findings IV.1: Stakeholders can see value and benefit in AI4SABI (6/7); current need for data-driven feedback exists (5/7)

Changing Mindsets

Although the benefits and value of using data-driven approaches are being acknowledged by increasingly more people, including our interviewees, many others (e.g. colleagues of the interviewees) are rather skeptical and have yet to be convinced. Precisely this is very decisive as the product owner of Subproduct A₁ states: "in order to enable data-driven approaches, the entire community would have to support and stand behind the idea and then assign a certain priority to this topic". In addition to that, the product manager of Product A clarifies that "all areas of the ecosystem would be affected" by the implementation of data-driven concepts. However, according to the product owner of Subproduct A₁ it "is difficult to gain acceptance of the new ways of working". The demand manager of Product B notes that "people need to be motivated to want to use it". Additionally, the product owner of Product B emphasizes that this will require a change of mindset for people to get used to it. Moreover, three of our interview partners stress that it is particularly challenging to convince management of the value of data-driven approaches because it usually requires additional resources to process and analyze data, build and maintain an infrastructure around the data, and prepare and visualize the results.

Findings IV.2: Challenging to convince others of value (4/7); change of mindset required (3/7); difficult to convince management to provide resources (3/7)

Priorities & Resources

According to six of the interviewees, the priority of introducing data-driven practices is still quite low partly due to the management’s skepticism. The product owner of Subproduct A₁ states that "at the moment this kind of topic does not get any priority due to the pressure of constantly developing new features". Related to this, the demand manager of Product B explains that it is a "challenge to integrate it in the timeline as the time-to-market expectations are very short". As a consequence of the low priority assigned to data-driven approaches, the interviewees feel that the available resources are not sufficient to drive the topic forward. Without resources to develop some initial solutions, it is difficult to demonstrate
and prove the value. However, only by clearly conveying the value and benefits of data-driven practices, the priority of these topics will be increased.

| Findings IV.3: Low priority due to pressure of developing new features (5/7); Low priority due to management’s scepticism (3/7); not enough resources provided to drive topic forward (3/7) |

7.4.2 Group II - Building AI4SABI Solutions

In order to investigate an additional perspective of why data-driven techniques like AI4SABI are only rarely successfully realized in practice, the second group of our interview study consists of roles responsible for implementing and building these types of solutions, specifically two software architects, one product owner (for operations), one data scientist, and one operations engineer. They are all experienced in developing and providing AI4SABI solutions to software engineering stakeholders, either by specifying requirements and analysis goals, collecting, processing and analyzing data, or building infrastructure to automate these processes. Though the interview guides are slightly tailored to the respective roles, each interview consists of the following elements: 1) generic processes and approaches when building or implementing AI4SABI solutions; 2) organizational- and people-related challenges; and 3) technical challenges.

Data Model, Processing & Quality

In order to obtain valuable insights for stakeholders, it is often unavoidable to combine data from different sources. (e.g. usage data, log data, sales data, performance data, bug reports etc.). Therefore, a number of different stakeholders who provide the data must be involved. The data scientist of Product A explains that this often "leads to different views on the data which can cause a lot of confusion". For that reason, according to the operations engineer of Product A, it is very important to define a unified data model that is applicable to all data sources in order to allow an easy mapping. As the amount of data is usually quite large (100GB per day in Product B), the software architect of Product B is responsible for decompressing the data, removing data that is not needed and store it in a usable format. Furthermore, the operations engineer of Product A notes that due to the variety of data sources, the data cleaning process is very multifaceted, and hence "becomes very complicated very fast". When automating the data collection it is crucial "to check and if necessary enforce consistency". Analogously, the software architect of Product B notes that it is critical to "make sure pipeline elements that are doing the preprocessing are always up and triggered at appropriate times, make sure the transformation tasks are working properly, [...] and having retry mechanisms in place in case they are failing".

An additional challenge mentioned by the software architect working as a consultant for multiple products, including Product B, is an insufficient quality of data provided by stakeholders. He explains that "due to low quality of data, the results [of the
analysis] are sometimes not good enough". As a result, he needs to "convey to the team that the quality of data needs to improve" even if that requires additional effort on the stakeholders sides. Otherwise the analysis will no longer be taken up and will consequently get stuck in a prototypical stage.

Findings IV.4: Challenge of combining multiple sources into one data model (4/5); ensure automated pipelines running correctly (3/5); ensure sufficient quality of data (3/5)

Mindset & Cultural Gap

Even after results are generated and presented to stakeholders, the acceptance of and believe in results is quite low. According to the product owner for operations of Product A the "results of analyses sometimes do not correspond with their beliefs" and they "would rather decide themselves what’s good and bad".

Both, the software architect of Product B and the software architect working as a consultant, describe it as a problem of mindset because on the one side "people are still skeptical and don’t see the benefit yet", and one the other side "to improve the quality of data [required for a sufficient quality and usefulness of results], process changes in their day-to-day work are inevitable".

The operations engineer of Product A believes that there is some kind of psychological effect because the stakeholders often "feel like an analysis generates additional effort and expense, so it hurts more than it helps". In addition to that, the product owner of Product A states that stakeholders often feel "pressure to explain oneself instead of perceiving it as something useful". Due to this, stakeholders often have a negative attitude towards analyzing software engineering data and "feel like data is being used against them", according to the data scientist of Product A.

Another big challenge is the interpretability and understandability of analysis results. The data scientist of Product A states that stakeholders often do not have any "motivation to engage with something if they don’t understand the technical side". For this reason, it is very "important to take the time to explain the analysis and results". However, in order to convey the technical knowledge necessary to convince stakeholders of the value of an analysis, some form of translation between roles is required. The data scientist brings up the example that even AI experts will not be able to communicate the required knowledge if they cannot express it in a way that is understandable to non-data-science stakeholders. Furthermore, it is important to consider that people are different. For instance, some people are very structured while others are more creative. As a result, different kinds of characters may need different kinds of support to take up AI4SABI. This should be taken into account when explaining an analysis and its results.

The operations engineer of Product A reports that even stakeholders who recognize the value in analyzing their data, find it difficult to specify their information needs and "to come up with any ideas themselves, they simply lack the time to think about it". Furthermore, the software architect of Product B notes that "people who are showing some interest, have to prove the value of the analysis to the 'higher' stake-
holders who are making the decisions on financing such analyses". Consequently, the
AI4SABI users do not only need to understand it, they also need to be able to defend
or even explain it successfully to their higher stakeholders to succeed. This can be
very challenging as the software architect working as a consultant explains: "some
managers are very skeptical and it can be difficult to convince them that it’s useful
or worth investing time or people".

| Findings IV.5: Impression by providers that acceptance and belief in usefulness
by users of AI4SABI is still low (5/5); Users’ perception that it hurts more than it
helps (3/5); explanations and translation very important to increase understand-
ability and interpretability (3/5); difficulties in specifying use cases for analysis
(2/5); challenging to convince others of value (2/5) |

Priority, Time & Resources

The aforementioned attitude towards AI4SABI and other data-driven approaches has
a direct impact on the priority, time and resources invested in driving these topics
forward. The software architect working as a consultant explains that stakeholders
are often "so much involved in the day-to-day activities that they don’t have time
to evaluate or try out something innovative". The data scientist of Product A notes
that "it’s becoming more present in managers’ minds but they lack of understanding
how to deal with the results".

Moreover, the software architect of Product B states that "the 'higher' stakeholders
often have different priorities, stabilizing the product for example". For that reason,
the software architect working as a consultant tries to keep the effort on the stake-
holders’ side as low as possible, although this is often difficult to achieve when it
comes to data access and improvement of data quality. Oftentimes, analyzing data
for one stakeholder requires certain actions from other stakeholders who do not di-
rectly benefit from the results and are, therefore, reluctant to spend their time on
such topics.

Ultimately, it is very important to get "commitment from management" in order to
assign a certain priority and the required resources to the topic.

| Findings IV.6: Low priority assigned by management (3/5); stakeholders too
occupied with their daily work (3/5); additional workload for stakeholders (3/5); |

7.5 Survey Results

The survey was sent to 1596 individual stakeholders working in software engineering,
230 (14%) of whom completed the questionnaire. Participants in the survey include
both technical (60%) and management roles (53%), whereas each person can also
be assigned to both roles. Table 7.3 presents a summary of the sample size, the
respondents and their level of experience. In addition to that, the different types of
tasks carried out by the respondents are listed in Table 7.4.
Table 7.3: Overview of sample size, respondents and experience levels

<table>
<thead>
<tr>
<th>Experience levels in current role (in years)</th>
<th>No. of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1 years</td>
<td>1596</td>
</tr>
<tr>
<td>&gt;1-3 years</td>
<td>230</td>
</tr>
<tr>
<td>&gt;3-5 years</td>
<td>159</td>
</tr>
<tr>
<td>&gt;5-10 years</td>
<td>138</td>
</tr>
<tr>
<td>&gt;10-15 years</td>
<td>657</td>
</tr>
<tr>
<td>&gt;15 years</td>
<td>121</td>
</tr>
</tbody>
</table>

Table 7.4: Tasks performed by the participants

<table>
<thead>
<tr>
<th>Business analysis</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept development</td>
<td>73%</td>
</tr>
<tr>
<td>Feature definition</td>
<td>61%</td>
</tr>
<tr>
<td>Staff and budget management</td>
<td>39%</td>
</tr>
<tr>
<td>Communication with customers</td>
<td>57%</td>
</tr>
<tr>
<td>Development / Implementation</td>
<td>61%</td>
</tr>
<tr>
<td>Infrastructure setup</td>
<td>26%</td>
</tr>
<tr>
<td>Database setup and/or maintenance</td>
<td>11%</td>
</tr>
<tr>
<td>Operations</td>
<td>19%</td>
</tr>
<tr>
<td>Data analysis</td>
<td>30%</td>
</tr>
</tbody>
</table>

7.5.1 Current Status in Projects

When asked about the current status of collecting and analyzing (usage and/or runtime) data in the respondents’ projects, 45% report that they are already collecting data, whereas 19% state that data is being collected but not yet analyzed. In the near future, an additional 19% plan to collect and analyze their data. The most common type of data being collected is log data (59%), followed by usage data (54%) and incident and bug reports (53%). An overview of the types of data being collected is given in Figure 7-2.

Figure 7-2: Types of collected data sources

Oftentimes, off-the-shelf tools are used to analyze and visualize collected data in the
participant’s projects. In particular, 52% of the respondents use tools like Microsoft Power BI, Tableau, Azure Log Analytics or Application Insights to extract some first information and visualize their data. Out of the 52% who are using these tools, 54% are either satisfied (52%) or very satisfied (2%) with the preparation and presentation of data and analysis results by the tools. Another 31% are neither satisfied nor dissatisfied, while 12% of users are dissatisfied and 3% very dissatisfied. In addition to that, we asked the participants who are using off-the-shelf tools how well they are able to interpret the analysis results provided by the tools. Only one third of the respondents claims that they are able to interpret the results very (32%) or extremely (4%) well, while 64% of the respondents state that they are only moderately (49%), slightly (11%) or not at all (4%) able to interpret the results. Furthermore, the respondents are asked about types of information or insights they wish for but are not able to retrieve from the tools they are currently using. The majority of comments are related to additional data sources that respondents would like to include in analyses and visualizations. However, a significant number of participants also mention the application and use of AI techniques such as pattern detection and prediction models.

Findings V.1: The majority of respondents is open towards using AI4SABI, many already apply data-driven approaches; many use off-the-shelf analytics tools, though some struggle to interpret the results; the type of additional customized information respondents ask for underpins the usefulness of AI4SABI

7.5.2 Challenges and Experiences in AI4SABI

In the third part of the questionnaire, respondents are asked to share their experiences with AI4SABI and the challenges they are facing during implementation or usage. In 47% of cases the respondents themselves and in 37% of cases colleagues or team members of the participants are applying basic analytic techniques (e.g. simple statistics) to manually analyze data. Only 18% of respondents report that they are successfully applying advanced analytics (e.g. complex statistics, machine learning etc.) in their projects. Furthermore, the participants are asked to rate the frequency of using these analyses for decision-making in practice. A total of 37% of participants respond that they either often (34%) or almost always (3%) use such analyses for decision making, while 36% use it sometimes and 27% either seldom (23%) or never (4%). In addition to that, we ask the respondents how difficult they experience the transformation of a prototypical analysis into a productized (automated & deployed) analysis. Only 7% assess it as not at all (2%) or slightly (5%) difficult, while 48% experience it as moderately difficult and 45% as either very (37%) or extremely (8%) difficult. Based on this, the participants are asked how they experience the effort of setting up an infrastructure for continuous data analysis. 14% of the participant rate the effort as very high and 37% as high. 39% have experienced an average amount of effort and only 9% claim the effort to be either low (8%) or very low (1%). A structured overview of the results is presented in Table 7.5.
Using results for decision making

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>Seldom</th>
<th>Sometimes</th>
<th>Often</th>
<th>Almost always</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4%</td>
<td>23%</td>
<td>36%</td>
<td>34%</td>
<td>3%</td>
</tr>
</tbody>
</table>

Difficulty of transforming a prototypical analysis into a productized (automated & deployed) analysis?

<table>
<thead>
<tr>
<th></th>
<th>Not at all</th>
<th>Slightly</th>
<th>Moderately</th>
<th>Very</th>
<th>Extremely</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2%</td>
<td>5%</td>
<td>48%</td>
<td>37%</td>
<td>8%</td>
</tr>
</tbody>
</table>

Effort of setting up an infrastructure for AI4SABI

<table>
<thead>
<tr>
<th></th>
<th>Very low</th>
<th>Low</th>
<th>Average</th>
<th>High</th>
<th>Very high</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1%</td>
<td>8%</td>
<td>39%</td>
<td>37%</td>
<td>14%</td>
</tr>
</tbody>
</table>

Table 7.5: Using results for decision making, productizing analysis and setting up an infrastructure

<table>
<thead>
<tr>
<th></th>
<th>Not at all</th>
<th>Slightly</th>
<th>Moderately</th>
<th>Very</th>
<th>Extremely</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of time and priorities</td>
<td>3%</td>
<td>9%</td>
<td>20%</td>
<td>53%</td>
<td>14%</td>
</tr>
<tr>
<td>Data privacy and security concerns</td>
<td>12%</td>
<td>21%</td>
<td>27%</td>
<td>28%</td>
<td>11%</td>
</tr>
<tr>
<td>Lack of knowledge of data sources</td>
<td>12%</td>
<td>23%</td>
<td>34%</td>
<td>23%</td>
<td>8%</td>
</tr>
<tr>
<td>within projects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of expertise in building</td>
<td>6%</td>
<td>21%</td>
<td>30%</td>
<td>32%</td>
<td>11%</td>
</tr>
<tr>
<td>infrastructure for data analytics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of expertise in data engineering</td>
<td>9%</td>
<td>19%</td>
<td>30%</td>
<td>36%</td>
<td>6%</td>
</tr>
<tr>
<td>(i.e. data collection, data processing)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difficulty of accessing, integrating</td>
<td>4%</td>
<td>18%</td>
<td>30%</td>
<td>28%</td>
<td>14%</td>
</tr>
<tr>
<td>and processing data from different</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of expertise in data analytics</td>
<td>8%</td>
<td>20%</td>
<td>30%</td>
<td>34%</td>
<td>7%</td>
</tr>
<tr>
<td>(i.a. selecting and applying</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>algorithms)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of data quality</td>
<td>8%</td>
<td>18%</td>
<td>33%</td>
<td>27%</td>
<td>14%</td>
</tr>
<tr>
<td>Effort of transforming a prototypical</td>
<td>4%</td>
<td>6%</td>
<td>31%</td>
<td>48%</td>
<td>11%</td>
</tr>
<tr>
<td>analysis to an automated and</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>continuous analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difficulties in interpreting results</td>
<td>9%</td>
<td>25%</td>
<td>40%</td>
<td>17%</td>
<td>2%</td>
</tr>
<tr>
<td>Lack of trust in results (either due</td>
<td>16%</td>
<td>36%</td>
<td>28%</td>
<td>17%</td>
<td>3%</td>
</tr>
<tr>
<td>to or independent of</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interpretability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of belief in usefulness of</td>
<td>26%</td>
<td>24%</td>
<td>33%</td>
<td>15%</td>
<td>2%</td>
</tr>
<tr>
<td>results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.6: Rating of adverse factors on the application of data analytics in the respondents’ projects; the bold numbers indicate which factors were ranked as either very (at least 30%) or extremely (at least 10%) influential by the respondents

In order to identify and understand the factors that negatively influence a successful application and usage of AI4SABI, we extracted a list of factors from existing literature and the interviews presented in Section 7.4. As part of the survey, the respondents are asked to rank how much these factors negatively impact the application and usage of AI4SABI in their projects on a 5-point Likert scale. There is always the possibility to skip and not rate a factor if participants consider that factor to be out of their area of expertise. Table 7.6 gives an overview of the factors and their rating.

A total of 67% of the survey’s participants perceive the lack of time and priorities as an either very (53%) or extremely (14%) important factor that impedes the implementation and usage of AI4SABI in practice. Furthermore, a lack of expertise in data analytics (41%), data engineering (42%) and in building infrastructure for data analytics (43%) is considered either very or extremely challenging by the respondents. The difficulty of accessing, integrating and processing data from different sources is
also rated either very or extremely high by 42% of the participants. The same applies to the effort of transforming a prototypical analysis into an automated and continuous analysis (59%). The plot in Figure 7-3 presents an additional illustration of the results. The negative impact of the individual factors is visualized from light (not at all) to dark (extremely). The larger the dark fraction in the bars is, the greater is also the negative influence of the respective factor.

Figure 7-3: Rating of adverse factors on the application of data analytics in the respondents’ projects

To conclude the survey, we offer a free text field for the respondents to share further experiences and challenges that impede a successful application and use of AI4SABI. Some participants consider it particularly difficult to change the mindset of others regarding AI4SABI and data-driven practices in general. One respondent claims that "the main problem is the people not being convinced that using data makes sense". Another one states "the concrete mindset is often missing. Only a training course is not sufficient". Related to this, one respondent feels like there is a "lack of management support, which is partly because it is difficult to provide 'use cases' that a manager will find valuable for the organization". For that reason, it is "difficult to continuously get the budget and staffing". From the opposite viewpoint a participant explains: "I am limited by cost savings at present, so it requires a step by step approach". Even "if time is spent on analysis, oftentimes there’s considerably less (sometimes: no) time to act on its result".

Moreover, the specification of use cases and the collaboration of different roles to achieve this can be very challenging. One respondent states that "data analysts answer questions, but [...] do not proactively provide ideas". Another one mentions that there is a "cultural difference between domain experts and data scientists".

152
Table 7.7: Key drivers and challenges derived from interview study (I) and survey (S)

From a technical perspective respondents claim that it is "difficult to get access to data from all the individual databases and sources" and ensuring data quality at the same time. Another participant adds that "prototypes are cheap, but the next step takes time and effort" because "models need validation, maintenance and automation" which consequently requires "setting up the right infrastructure" for it.

Findings V.2: Results are sometimes used for decision making; the effort to set up an infrastructure is perceived as high; the transformation from prototypes to continuous & automated analysis is difficult; it is difficult to access, integrate and process data from different sources; Critical adverse factors: lack of time and priorities, lack of expertise in data engineering, data analytics & building infrastructure, and lack of data quality.

7.6 Identification of Challenges and Key Drivers

This section aims at addressing RQ1: What are the key drivers for analyses not being productized after the prototypical stage? To achieve this, we analyze and compare the results of the interview study and the survey. We were able to identify multiple interdependent challenges that often prevent AI4SABI and other data-driven approaches...
from being successfully applied and used in practice. We observed that several of these challenges act like drivers that impact and partially amplify each other, effectively producing a vicious circle. Table 7.7 gives an overview of the identified challenges and key drivers, and indicates the source (interview study or survey) of each challenge. While there are additional challenges and factors potentially influencing a successful application of AI4SABI (e.g. data privacy and security concerns), we only selected interrelated challenges that are contributing to the key drivers of the vicious circle. A description of these drivers is provided in the following subsections before the vicious circle is explained and presented.

7.6.1 Lack of Priority, Time and Resources

In order to successfully implement and use AI4SABI, a certain priority and consequently time and resources need to be assigned to the topic. The majority of participants in our study report that the priority assigned by the management is insufficient for driving these topics. Therefore, not enough resources are provided to build new or improve existing AI4SABI solutions.

7.6.2 Low Data Quality

On multiple occasions, interviewees and respondents in our study mention the lack of data quality in their projects. In addition to that, the data sources in the context of AI4SABI are often distributed and spread across multiple teams. In order to extract valuable insights, the data sources need to be combined. This often leads to the challenge of storing and integrating data from various sources in a consistent and accessible way. As a consequence, the data are often prone to be of lower quality. In addition to that, the lack of priority, time and resources impedes the improvement of data quality.

7.6.3 Inability to Cross Cultural Gap

Based on the available data, possible use cases and topics of interest for an analysis need to be identified and discussed. In this context, we encountered a conflict during the evaluation of results. On the one hand, providers of AI4SABI claim that it is difficult to obtain requirements or specifications from the stakeholders who use the analysis in the future. On the other side, AI4SABI users feel overwhelmed by this task and, for that reason, expect the data analysts to provide inspiration and concrete suggestions. As a result, both stakeholders feel frustrated and not well understood by the other. To solve this, more time would need to be dedicated to more intensive exchanges and discussions between both parties. Data analysts need to better understand the stakeholders’ domain-specific needs, while stakeholders need to acquire a technical understanding of the applied analyses and their results (e.g. by participating in trainings). However, as mentioned by one of our interviewees "single introduction sessions will not be sufficient", but a willingness to engage in a long-term collaboration will be required.
7.6.4 Ineffective Prototypical Analysis

Eventually, AI4SABI providers start with a prototypical implementation of a given use case. Mostly these initial implementations are based on data snapshots to produce some first results. However, in order to actually make use of the results, an analysis needs to be based on high-quality data and it should be executed in a continuous and automated way. The findings of our case study indicate that the results are often not good enough due to a low quality of data. Additionally, it is complex, takes a lot of effort, and requires expertise to automate analyses. This leads to two challenges: On the one side, users of AI4SABI may not be able to recognize the value of an analysis and its results. On the other side, additional resources need to be acquired to realize the productization of the analysis.

7.6.5 Inability to Prove Value

Finally, before required resources can be allocated, the value of AI4SABI or other data-driven practices needs to be proven to relevant stakeholders. That includes not only the users of AI4SABI as primary stakeholders but also indirect stakeholders such as managers or other affected colleagues. This turns out to be a difficult task because, especially in the short-term, the advantages and benefits of AI4SABI are not directly perceived by customers. As a result, the pressure of implementing new features is usually higher than analyzing existing ones. Therefore, the former will in many cases be prioritized by the management.

In order to prove the value, oftentimes a change of mindset is required since people need to get used to a new way of working. It is important to establish trust and support stakeholders in understanding and interpreting the results. Otherwise, they are going to lose interest.

7.6.6 Vicious Circle

In summary, the low priority assigned to the topic results in two consequences: On the one side, an inadequate amount of time is spent on dealing with data quality. On the other side, the discussions between users and providers of AI4SABI are not sufficient to overcome communication problems, create a mutual understanding and define relevant use cases. This cultural gap, along with poor quality of data, leads to a prototypical implementations whose results cannot deliver the desired value. In turn, if the value is not clearly visible, it is challenging to convince others of that value. So the vicious circle (see Figure 7-4) is complete as this again will result in not increasing the priority of the topic.

7.7 Identification of Potential Solutions

This section aims at addressing RQ2: *How can the identified challenges and key drivers be addressed?* Based on they key drivers presented in the vicious circle, we review a list of key publications as well as the conducted interviews and survey for
potential solutions that can support practitioners in taking a targeted approach at breaking the circle.

### 7.7.1 Solutions Proposed in Literature

One of the key challenges we identified in our study was a *low quality of data* in the interviewees’ and survey respondents’ projects. The heterogeneity of data and tools makes it difficult to reuse work, including cleaned or preprocessed data, of others. To address this, a large-scale survey among data scientists revealed that centralizing data and the corresponding definition as well as agreeing on and using only a subset of available tools can decrease the engineering effort across teams and lead to a better reusability of components [Kim et al., 2017].

Furthermore, data scientists at Microsoft use qualitative channels for the validation of quantitative data to ensure the meaningfulness of measurements [Kim et al., 2016]. This can, for instance, be done by surveying a subset of people who are responsible for the data (e.g. users or customers, if feasible) [Kim et al., 2016] or by involving experts or stakeholders who are familiar with the data [Kim et al., 2017], [Passi and Jackson, 2018]. A triangulation of multiple data sources increases the confidence in results [Kim et al., 2016].

To overcome the *inability to cross the cultural gap* between data scientists and other stakeholders, a close and continuous interaction and collaboration is crucial [Kim et al., 2016]. Moreover, stakeholder questions should be defined early on and refined in an iterative manner and data scientists should support the stakeholders in inter-
interpreting the data as well as the results [Kim et al., 2016]. On the other side, data scientists should also consider and try to understand the business perspective since each analysis should be based on a set of predefined goals and business decisions that are to be supported by the analysis [Kim et al., 2017], [Provost and Fawcett, 2013]. By indicating how the business perspective is represented in the data and the model, an analysis can become more intuitive to stakeholders and the perceived usefulness of an analysis increases [Passi and Jackson, 2018].

Oftentimes, one-time analyses on snapshots of data are ineffective to show the desired value of an analysis. Therefore, it is important to "go the last mile" [Kim et al., 2016] and operationalize the analyses. In a case study at Microsoft, one of the interviewees explains that he or she spends 50% of the time on an analysis and the same amount of time on the integration into the product [Kim et al., 2016].

In order to clearly underline the value and benefit of an analysis, it is important to define and discuss concrete actions that can be taken based on the results [Kim et al., 2016]. Moreover, explanations of analytics results should be kept as simple as possible to increase the understandability of the results and to show what stakeholders can achieve by using it [Kim et al., 2016], [Passi and Jackson, 2018]. After explaining an analysis, stakeholders should feel like they can use it by themselves without the need of constantly having a data scientist by their side [Kim et al., 2016].

To achieve this, data scientists should not use data science measures to explain the results, but rather translate their findings into the stakeholder's domain (e.g. "how much money can be saved?" or "how many customer calls can be prevented?") [Kim et al., 2016]. It is important to consider that "stakeholders' trust in data science systems stems not only from model results and performance metrics, but also from some explanation or confidence in a model's inner working" [Passi and Jackson, 2018]. Therefore, additional, stakeholder-targeted explanations minimize the risk of stakeholders misinterpreting a data science measure (e.g. accuracy score, precision, recall etc.) and build trust in results by explaining and visualizing results in a way that non-data-scientists can understand (e.g. showing probabilities of certain events instead) [Passi and Jackson, 2018]. Thereby, explanations should be continuously evaluated and improved in collaboration with the stakeholders [Dam et al., 2018], [Passi and Jackson, 2018].

Lastly, we were not able to find any solutions that address the last challenge in the vicious circle (lack of priority, time and resources) as the costs and financing of AI-based software analytics or business intelligence solutions are being neglected in existing literature.

7.7.2 Solutions Proposed in Interviews

Related to the challenge of low data quality the operations engineer of Product A explains that they are using a "unified data model based on days to keep the effort to a minimum". Since all tables have the same granularity, they are easily extensible and connectible which can save a lot of time later on. In addition to that, the software architect working as a consultant for Product B explains that it is important to "identify which data sources are really important and to prioritize the data according
to the stakeholders’ needs”.
In order to overcome the inability to cross the cultural gap between data scientist and other stakeholders, the data scientist of Product A highlights that while introduction sessions to machine learning are very important to help in understanding how the analyses work and how results can be interpreted, "single introduction sessions will not be sufficient". Therefore, shorter and more importantly iterative and continuous collaboration sessions are recommended. This is also stressed by the product owner and operations engineer of Product A. In order to provide meaningful AI4SABI solutions it is crucial to continuously work with the stakeholders, collect their feedback, and identify the problems they are facing. Otherwise, the stakeholders will stop trusting the analysis and will no longer want to use the AI4SABI solution.
The inability to prove value can be overcome by conducting sessions with the stakeholders to present and explain the results. Specifically, the software architect working as a consultant for Product B highlights the importance of explaining what kind of input data is used and what type of results or insights can be derived from the analysis. In addition to that, the software architect of Product B explains that they think about the presentation and visualization of data and information in the very beginning and even already define the UI elements as the visualization of results often supports stakeholders in comprehending what the results say and how they can use them. Moreover, the data scientist of Product A stresses that "trainings should not be on a high level but on a level that non-data-scientists can understand". Adding to that, the product owner for operations of Product A proposes to conduct role-specific trainings because he feels like "only few product managers understand machine learning" and he also emphasizes that "people need to understand the benefit and value of such analyses, otherwise even the coolest stuff is not worth anything".
For the remaining challenges ineffective prototypical analysis and lack of priority time and resources we were not able to extract any solutions out of the interviews.

7.7.3 Solutions Proposed in Survey

In regard to the low quality of data, one of the participants in the survey mentions that they are "enriching the current data source with more relevant data to get useful results". If the quality of data of one data source is insufficient, it can be helpful to include additional data sources.
Furthermore, the challenge of ineffective prototypical analysis is addressed by two of the survey’s participants. One of the participants states that "AI is very powerful, but it seems it just gets applied to any problem case in order to solve it". Therefore, the respondent proposes to only apply AI to use cases where it really makes sense and to prioritize the use cases accordingly. Another survey participant notes that: "Currently, if time is spent on analysis, oftentimes there’s considerably less (sometimes: no) time to act on its result. If one acts on an instinct rather than profound analysis, there’s more time to act and a better chance to actually have any result". Therefore, the respondent states that a crucial change is required to "allow time for both analysis and to act on its result". Moreover, the same participants also highlights the importance of prioritizing AI4SABI projects before starting and finalizing
started projects.
Related to the lack of priority, time, and resources one of the survey’s respondents explains that he or she is "trying to push this inside the team but due to time constrains nothing is taken into the backlog". Therefore, they are hiring students who can implement some first use cases in a cost-effective way. In addition to that, two further participants note that it is helpful to clearly show and discuss the tradeoff between the cost of the analysis and the value or business impact that it can provide. Knowing and understanding this tradeoff can make it easier for managers or decision makers to weigh up the decision on whether to allocate more time and resources for this topic.

We were not able to extract any solutions out of the survey that could help in overcoming the inability to cross cultural gap and the inability to prove value.

7.7.4 Categorization and Validation of Solutions
In order to structure and validate our findings, we assign the identified solutions to the key drivers in the vicious circle, then group the solutions into categories and finally discuss the categorized solutions with a group of experts who also rate the solutions’ usefulness on a scale from one to five.

Solution Categorization

Low data quality For the first key driver we were able to identify six solutions that can either directly (e.g. through validation) or indirectly (e.g. via more effective usage or processing of data) help to improve data quality (see Table 7.8).

<table>
<thead>
<tr>
<th>Solution</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized storage of data &amp; definition</td>
<td>Data storage &amp; model definition</td>
</tr>
<tr>
<td>Unified data model (e.g. based on days) for same granularity of all tables → easily extensible and connectible</td>
<td>Data storage &amp; model definition</td>
</tr>
<tr>
<td>Agree on a subset of tools for data storage &amp; processing</td>
<td>Data storage &amp; model definition</td>
</tr>
<tr>
<td>Triangulation of multiple data sources</td>
<td>Data validation</td>
</tr>
<tr>
<td>Enhancing data with additional relevant data sources</td>
<td>Data validation</td>
</tr>
<tr>
<td>Use of qualitative channels for validation, e.g. involve stakeholders</td>
<td>Data validation</td>
</tr>
</tbody>
</table>

Table 7.8: Identified solutions for key driver: Low data quality

The indirect solutions include a centralized storage of data and the corresponding definition, a unified data model for the same granularity of all tables as well as the agreement on a subset of available tools for data storage and processing. These solutions are not directly related to the quality of data but they can indirectly contribute to it by making the access, usage, and processing of data more effective which leads to more time that can be spent on dealing with data quality. Since the solutions all cover aspects that need to be considered at the beginning of collecting data, they are assigned to the category data storage & model definition.

The more direct methods to improve data quality are mostly related to data validation. This includes a triangulation of multiple data sources, an enrichment of data with additional relevant data sources, and the use of qualitative channels for validation.
Inability to cross cultural gap  The solutions for this key driver are mainly related to the collaboration between the providers of AI4SABI solutions and stakeholders who are going to use it (see Table 7.9).

<table>
<thead>
<tr>
<th>Solution</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close continuous collaboration with stakeholders [e.g. weekly data meetups]</td>
<td></td>
</tr>
<tr>
<td>Iterative definition and refinement of questions</td>
<td></td>
</tr>
<tr>
<td>Regular feedback sessions with stakeholders to support interpretation of data and results and identify problems faced by stakeholders</td>
<td></td>
</tr>
<tr>
<td>Introduction sessions are important but a single session will not be enough → rather multiple, short sessions on a regular basis instead of 1-day workshop</td>
<td></td>
</tr>
<tr>
<td>Clearly specify and define goals and business decisions show how this is represented in the data</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.9: Identified solutions for key driver: Inability to cross cultural gap

In order to cross the identified cultural gap, a continuous collaboration with stakeholders is crucial. This also includes an iterative definition and refinement of questions and regular feedback sessions to identify problems and support the stakeholders in understanding an analysis. Moreover, a sufficient amount of time should be taken to introduce stakeholders to the topic of AI4SABI as single introduction sessions will not be enough. On the other side, providers of AI4SABI need to become familiar with the goals and business decisions that are to be supported by their analyses. Showing the stakeholders how the data and models are linked to the business perspective will help them in building up trust in an analysis.

Ineffective prototypical analysis  In order to overcome the ineffectiveness of prototypical analyses, practitioners highlight the importance of planning and taking time apart from implementing and running an analysis even if that leads to fewer or more simple analyses (which are less time-consuming). Taking the time for both operationalization and acting on the results, is going to make analyses more effective by being more actionable and, therefore, also valuable for stakeholders. Moreover, the usefulness of applying AI to an use case should be carefully evaluated prior to implementation. Only use cases where it really makes sense to apply AI should be taken up and these use cases should be prioritized accordingly. The proposed solutions are also presented in Table 7.10 and can all be assigned to the category planning.

<table>
<thead>
<tr>
<th>Solution</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan fewer or less time-consuming analyses to have time for operationalization</td>
<td></td>
</tr>
<tr>
<td>Allow time for both analysis and acting on its results</td>
<td></td>
</tr>
<tr>
<td>Only apply AI to use cases where it really makes sense &amp; prioritize use cases</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.10: Identified solutions for key driver: Ineffective prototypical analysis

Inability to prove value  The value of an analysis can be clearly conveyed to stakeholders by providing explanations and translations of an analysis’ result (see Table 7.11).

On the one side, regular sessions should be conducted to present and explain the results while the explanations should be kept simple so stakeholders feel like they can
use it by themselves. It is important to discuss and explain what kind of actions can be taken based on the results and to think about how to present and visualize the results early on as part of the implementation. On the other side, results should be translated to the stakeholder’s domain. So instead of using data science measures other domain-specific indicators should be communicated (e.g. "what is the business value?" or "how much time can be saved?"). This also applies to trainings that are conducted with stakeholders. These should not be on a high level but on a level that non-data-scientists can understand, maybe even role-specific.

Lack of priority, time and resources  Due to the pressure of their every day work, the management’s priority of spending time and resources on analyses that do not directly go into the product is often quite low, especially if the value of that analysis is not clearly visible. In order to keep the costs in the beginning to a minimum, students can be hired as a cost-effective way to implement some first use cases. Moreover, it can help to clearly show and discuss the tradeoff between the costs that will occur and the value that an analysis could provide. Both solutions are assigned to the category costs (see Table 7.12).

Table 7.12: Identified solutions for key driver: Lack of priority, time and resources

Solution Validation

In order to validate the impact and usefulness of the solutions, we conduct expert interviews and ask the participants to discuss and rate the usefulness of solutions on a scale from one to five. Four of the five experts also participated in the initial interview study. The fifth, who was not available then, was added due to his expertise in the area. An overview of the solution categories, their sources as well as their expert rating is provided in Figure 7-5. The solutions are grouped by their respective categories which are assigned to the key drivers of the vicious circle. The boxes next to the categories indicate the sources of the solutions (literature, interviews, and/or survey) and the average rating of the experts who participated in our study.
Data Storage & Model Definition  The first solution category comprises solutions proposed in the literature and the interviews. Especially the unified data model is perceived as very useful to make the data more "transparent and easier to understand". Independently of each other, all experts rated the usefulness of the solution as 4 (on the scale from 1 to 5).

Data Validation  The solutions of the category data validation were derived from the literature as well as the survey and are discussed a bit more critically by the experts. The operations engineer of Product A explains that the triangulation of multiple data sources and the enrichment of additional data sources mostly increases the "confidence for the users, but not for the providers". The software architects of Product B agree that it is important to involve the stakeholders and take their domain knowledge into account. Overall, two experts of Product A gave this category
a rating of 2.0, while the remaining experts rated it a 3.0 resulting in an average rating of 2.7/5.

**Collaboration**  The third solution category *collaboration* consists of solutions extracted from the literature and interviews. It is perceived as very useful in theory but not really feasible in practice. The data scientist of *Product A* explains that "it’s a good idea but there is simply no time. Who is going to realize this?". The software architect of *Product B* states that "it depends on where the push is coming from". The communication within a team is usually much easier as compared to across-teams. Depending on the team setup of users and providers of AI4SABI, a close collaboration can become difficult. Moreover, the stakeholders’ willingness to invest time in such a close and time-consuming collaboration depends on the stakeholders’ personal interests, priorities, and motivation. Due to the difficulty of realizing this in practice, four out of six interviewees rate this solution category a 2.0 and two other interviewees a 3.0 and 4.0 respectively. This results in an overall expert rating of 2.5/5.

**Planning**  The solutions of the *planning* category are based on all the considered sources (literature, interviews, and survey) and are well received by the group of experts. The operations manager of *Product A* confirms that "it is better to have something simple than something very complex that is not complete". The software architect of *Product B* also agrees that this is "very relevant". In total, two out of six interviewees rate this solution category a 5.0 and the other two a 4.0. Ultimately, this results in an overall expert rating of 4.3/5.

**Explanation**  The solution category *explanation* comprises solutions derived from the literature as well as the interviews. The group of expert agrees that this is indeed a very important topic to consider. The data scientist of *Product A* states that "it is always good when stakeholders see some first visualizations early on". He adds that the information or numbers that are being reported "must be easy to understand". The software architects of *Product B* also consider the solutions of this category very useful. One of them explains that it also depends on the stakeholder. Some stakeholders really want to understand why a ML-model predicts something in a certain way while others only care about whether the results are useful or not and what the next actions are. Therefore, the explanations need to be adapted to the stakeholders’ personalities and needs. As a result, four out of six interviewees rate this category a 5.0 and two a 4.0 resulting in an overall rating of 4.7/5.

**Translation**  Very similar to the previous category, the solutions of the category *translation* were derived from the literature and the interviews and are perceived as very valuable by the group of experts. The data scientist of *Product A* confirms that these solutions are "very important". The software architects of *Product B* also state that they fully agree with the solutions and have "nothing to add". Two of the
interviewees give a rating of 5.0 and the remaining four interviewees rate this solution a 4.0. Consequently, this results in an overall rating of 4.3/5.

**Costs**  For the last solution category *costs* we were only able to identify solutions from one of the sources, specifically the survey. Regarding the proposal of hiring students to support the implementation of use cases, the operations engineer of *Product A* confirms that they are also doing that and that "it is good for a quick start to get something done at a lower cost". The software architect of *Product B* adds that while hiring students is cost-effective, it sometimes comes at the cost of quality. As a result, five out of six interviewees give a rating of 4.0 and one rates it a 3.5. Finally, this results in an overall rating of 3.9/5.

7.8  Threats to Validity

7.8.1  Internal Validity

One threat to the internal validity of our study is the limited number of products and the unequal distribution of interviewees among product teams. We compensated this by including the consultants who both work for *Product B* but also for a number of other products.

Moreover, it is likely that the respondents of our survey participated due to their interest in the topic of AI4SABI, thereby potentially causing a positive bias in the results. Due to the limitations of the survey tool, dependencies between questions and factors could not be taken into consideration.

Due to the lack of successfully deployed AI4SABI solutions, the group of experts interviewed for validating the solutions consists mostly of technical experts (providers) as compared to non-technical stakeholders (users). Therefore, we cannot rule out a slightly biased distortion of the results due to the over-represented providers of AI4SABI.

7.8.2  External Validity

The limited number of interviewees in the interview study might have an impact on the generalizability of the results which constitutes an external threat to validity. We aimed at compensating this by using a mixed-methods approach and additionally conducting a survey to answer the same research questions. However, the same applies to the expert interviews conducted to validate the potential solutions.

Furthermore, the transferability of results to other contexts might be affected by a positive bias in the survey results. While the overarching case study was conducted in multiple unrelated and independent product teams, they all originate within a similar industrial context which might impact the external validity of the study. However, with the many roles, organizational units and products represented in the study and by multi-source data collection
as the basis for our results, we believe that the findings we present provide value also outside the specific context of the case study company.

7.9 Conclusion

While the use of artificial intelligence for software analytics and business intelligence can in theory be a valuable asset to retrieve and extract meaningful information for decision-making, it is very difficult to realize it in practice, specifically in a continuous and automated manner.

In our study we have identified five interdependent key drivers that impede a meaningful utilization of AI4SABI: A lack of priority, time and resources results in a low quality of data and the inability to cross the cultural gap between data scientists and other stakeholders. This leads to an inefficient prototypical analysis and the inability to prove its value, which again prevents an increase of the priority and consequently also time and resources. As a result, these key drivers form a vicious circle that is difficult to break.

To remedy that, we have extracted and categorized a set of solutions out of existing literature, the interviews and the survey to support practitioners in taking a targeted approach at breaking the circle. The solutions are grouped into seven categories: Data storage & model definition, data validation, collaboration, planning, explanation, translation, and costs.

Indeed, the selection of solutions may depend on project-specific characteristics. In fact, the mentality and priorities of the specific stakeholders involved could act as a lever to select the solutions that are most promising. For instance, if a technical stakeholder such as a data engineer aims at driving the topic forward, he or she will could start by improving the data quality and data scientists could additionally try to work on or improve explanations provided to other stakeholders. On the other side, potential users of AI4SABI (e.g. product managers) could proactively promote the collaboration with data scientists and maybe even try to include colleagues that are still skeptical.

Currently, the validation of our study is limited to the experience and perception of the interviewed group of experts. In future work, we intend to conduct a long-term evaluation of applying the proposed solutions in practice.
Chapter 8

An End-to-End Framework for Productive Use of Machine Learning in Software Analytics and Business Intelligence Solutions


Abstract

Nowadays, machine learning (ML) is an integral component in a wide range of areas, including software analytics (SA) and business intelligence (BI). As a result, the interest in custom ML-based software analytics and business intelligence solutions is rising. In practice, however, such solutions often get stuck in a prototypical stage because setting up an infrastructure for deployment and maintenance is considered complex and time-consuming. For this reason, we aim at structuring the entire process and making it more transparent by deriving an end-to-end framework from existing literature for building and deploying ML-based software analytics and business intelligence solutions. The framework is structured in three iterative cycles representing different stages in a model’s lifecycle: prototyping, deployment, update. As a result, the framework specifically supports the transitions between these stages while also covering all important activities from data collection to retraining deployed ML models. To validate the applicability of the framework in practice, we compare it to and apply it in a real-world ML-based SA/BI solution.
8.1 Introduction

A vast amount of data is produced by software-intensive systems every day. As a result, software providers often try to gain insights from data using software analytics [Menzies and Zimmermann, 2013] or business intelligence [Negash and Gray, 2008] (SA/BI) tools. As existing tools are typically quite generic and can often not provide the desired depth of product-specific and stakeholder-targeted information, there is often a need for customized software analytics or business intelligence (SA/BI) solutions that leverage the full potential of modern machine learning (ML) techniques. However, as such solutions are used as internal systems for monitoring or decision-making, these are often not perceived as something of direct customer value by managers. This results in a lack of priority, time and, resources assigned to setup and maintain ML-based SA/BI solutions [Figalist et al., 2020]. In addition to that, the effort of going beyond a prototypical analysis and deploying it to and maintaining it in production is perceived as extremely high [Figalist et al., 2020], [Sculley et al., 2015]. Paired with a lack of expertise in this domain, which is often the case if the actual product is not related to ML [Buse and Zimmermann, 2012], custom ML-based SA/BI solutions are rarely deployed in production [Figalist et al., 2020]. Nevertheless, this is considered crucial in order to continuously gain valuable insights and use it for actual decision making [Lin and Koltz, 2012].

To address this, we conduct a literature review of important domains related to ML, specifically data management and processing, model building, and model deployment. The results are then used to derive a framework for building end-to-end ML-based SA/BI solutions consisting of three iterative cycles: a prototyping cycle, a deployment cycle, and an update cycle. To validate the applicability of the framework in practice, we compare it to and apply it in a real-world, customized ML-based SA/BI solutions.

The contribution of this paper is an end-to-end approach that covers all steps from data collection to retraining deployed ML models while at the same time taking the different conceptual stages into consideration (prototypical, deployment, and update). By specifically addressing the transition between these stages, our framework supports practitioners in advancing their prototypical analysis to a deployed and continuously retrained ML model.

The remainder of this paper is structured as follows: First, we outline the background of our study in Section 8.2. In Section 8.3 we provide an overview of the research method and the study design. The results of the literature review are presented in Section 8.4, before introducing the framework in Section 8.5. The framework validation is outlined in Section 8.6, followed by a conclusion in Section 8.7.

8.2 Background

The term software analytics (SA) describes analytics performed on software data to generate valuable insights for various stakeholders, from managers to software engineers, that ultimately support their decision making [Buse and Zimmermann, 2012].
Related to this, the field of business intelligence (BI), sometimes also referred to as business analytics, applies data mining techniques to operational data in order to derive high-quality information for managerial decision making [Negash and Gray, 2008].

With its increase in popularity, artificial intelligence soon became an integral part of SA and BI solutions [Chen et al., 2012], [Dam et al., 2018]. As a result, many companies aim at getting the most out of their data by running ML-based analyses on it. While some knowledge can be extracted using out-of-the-box tools, more in-depth analyses often require custom ML solutions.

In many cases, these custom solutions start out as a prototypical analysis or a proof of concept [Figalist et al., 2020], [Sculley et al., 2015]. However, in order to make actual use of the results, they need to be provided in a continuous manner by deploying the model to production and retraining the model on a regular basis [Lin and Kolcz, 2012]. Precisely this is the point at which custom ML-based SA/BI solutions often get stuck. In a previous study [Figalist et al., 2020], we identified a vicious circle that frequently prevents an end-to-end implementation of such analyses. One of the key issues is that an ineffective prototypical analysis can often not prove the value that it could deliver in production, leading to a lack of priority, time, and resources assigned to the topic [Figalist et al., 2020].

Moreover, in the context of SA and BI there is often a lack of expertise in data engineering, data analytics, and in building an infrastructure for both [Buse and Zimmermann, 2012], [Figalist et al., 2020]. For this reason, the framework presented in our study aims at compensating this to some extent by providing a structured approach for the transition between prototypical analysis and productively usable analysis.

### 8.3 Research Method & Study Design

As an end-to-end development of ML-based SA/BI solutions requires broad knowledge that is distributed across several, well-researched domains, we selected a deductive research approach for our study. Deductive approaches rely on existing theories for building hypotheses which are then confirmed or rejected using real-world observations [Runeson et al., 2012]. The overall research process is outlined in Figure 8-1.

![Figure 8-1: Research Process](image)

As a first step, we conducted a literature review [Keele, 2007] which serves as the foundation for our study. Based on the requirements of our framework, we identified
three overarching categories that comprise the results of our review: data management and processing, model building, and model deployment. To achieve our research goal, we queried common scientific libraries (IEEEExplore, ACM Digital Library, ScienceDirect, Springer Link) using search terms related to the respective categories: data (quality, cleaning, preprocessing, transformation, management, continuous extraction) and machine learning model (training, evaluation, deployment [pipeline], management, serving).

As inclusion criteria we defined 1) research papers that outline approaches and/or challenges in data management and processing, model building, or model deployment; and 2) case studies and experience reports describing concrete actions and processes for at least one of the categories. We excluded non-scientific contributions (e.g. posters or presentations/talks) and studies that were not written in English.

Next, we extracted all mentioned activities and challenges out of each selected paper and accumulated the results to common activities and challenges based on the frequency of occurrences. In order to derive a framework for productively applying ML in SA/BI solutions, we merged and systematically arranged the key activities of the investigated domains.

To validate the applicability of the framework in practice, we first compare it to the current state of a real-world ML-based SA/BI solution being developed for an industrial platform provider. In a second step, we utilize the framework to strategically plan and direct the upcoming activities. To achieve this, we collaborated with two software architects and two product managers of the platform. The product managers are the future user of the system and, therefore, provided us with a specific use case while the software architects supported us in building the ML-based SA/BI solution. The platform itself is based on Amazon Web Services¹ (AWS). For this reason, we utilize existing AWS services for implementing and deploying our solution. In order to get a comprehensive picture of all the activities, we interviewed the stakeholders in several recap sessions to gain a detailed understanding of individual steps that we could not directly be involved in due to company processes.

8.4 Literature Review

8.4.1 Data Management and Processing

The most important prerequisite for training accurate ML models is providing high-quality training data [Polyzotis et al., 2018], [Schelter et al., 2018]. At the same time, assembling high-quality data sets, and engineering and selecting appropriate features based on it, is very time-consuming and requires a vast amount of effort and resources [Domingos, 2012].

As a result, we investigate the common activities (see Table 8.1) in data management and data processing required for a successful application in machine learning systems as well as the challenges (see Table 8.2) that come with these activities. The identified activities can be grouped into six overarching categories: 1) Data preparation; 2) data

¹https://aws.amazon.com/
<table>
<thead>
<tr>
<th>Activity</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data preparation</strong></td>
<td></td>
</tr>
<tr>
<td>Identification of features and their properties based on raw data</td>
<td>[Domingos, 2012], [Lin and Kolez, 2012], [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>Transformation of input data to training data</td>
<td>[Baylor et al., 2017], [Breck et al., 2019], [Domingos, 2012], [Polyzotis et al., 2018], [Rajaram et al., 2020]</td>
</tr>
<tr>
<td><strong>Data cleaning</strong></td>
<td></td>
</tr>
<tr>
<td>Investigating and understanding effect of cleaning data on model accuracy &amp; filtering out uncorrelated data</td>
<td>[Cuzzocrea et al., 2011], [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>Ensure data quality, specification of (quality) rules &amp; actions for rules</td>
<td>[Baylor et al., 2017], [Chu et al., 2016], [Khayyat et al., 2015]</td>
</tr>
<tr>
<td>Detection of data errors</td>
<td>[Chu et al., 2016], [Khayyat et al., 2015]</td>
</tr>
<tr>
<td>Definition of data fixes &amp; execution of error repairs</td>
<td>[Chu et al., 2016], [Khayyat et al., 2015], [Polyzotis et al., 2018], [Volkovs et al., 2014]</td>
</tr>
<tr>
<td><strong>Data validation</strong></td>
<td></td>
</tr>
<tr>
<td>Triggering validation pipeline for each batch of data</td>
<td>[Breck et al., 2019], [Schelter et al., 2018], [Volkovs et al., 2014]</td>
</tr>
<tr>
<td>Generation of descriptive statistics of data, checking data properties based on specified schema/patterns &amp; identification of errors or anomalies in training data</td>
<td>[Baylor et al., 2017], [Breck et al., 2019], [Polyzotis et al., 2018], [Rajaram et al., 2020], [Schelter et al., 2018], [Volkovs et al., 2014]</td>
</tr>
<tr>
<td>Identification of features with significant impact on model accuracy</td>
<td>[Domingos, 2012], [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>Identification of dependencies to other data sources or infrastructure</td>
<td>[Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>Comparison of training and serving data to identify potential deviations</td>
<td>[Breck et al., 2019], [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td><strong>Data evaluation</strong></td>
<td></td>
</tr>
<tr>
<td>Performing sanity checks on data</td>
<td>[Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>Evaluation of choice and encoding of data based on model results</td>
<td>[Domingos, 2012], [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td><strong>Data serving</strong></td>
<td></td>
</tr>
<tr>
<td>Transformation of serving input data to serving data processible by model</td>
<td>[Baylor et al., 2017], [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>Channeling serving data back as training data</td>
<td>[Polyzotis et al., 2018]</td>
</tr>
<tr>
<td><strong>Extract, transform, load (ETL)</strong></td>
<td></td>
</tr>
<tr>
<td>Extraction of data from sources</td>
<td>[Dayal et al., 2009], [Vassiliadis, 2009], [Vassiliadis and Simitsis, 2009]</td>
</tr>
<tr>
<td>Transportation of data to processing pipeline (e.g. for data cleaning or filtering)</td>
<td>[Dayal et al., 2009], [Vassiliadis, 2009], [Vassiliadis and Simitsis, 2009]</td>
</tr>
<tr>
<td>Transformation of source data to target values</td>
<td>[Dayal et al., 2009], [Vassiliadis, 2009], [Vassiliadis and Simitsis, 2009]</td>
</tr>
<tr>
<td>Loading of cleaned &amp; transformed data</td>
<td>[Dayal et al., 2009], [Vassiliadis, 2009], [Vassiliadis and Simitsis, 2009]</td>
</tr>
</tbody>
</table>

Table 8.1: Common activities in data management and processing for machine learning
During the data preparation, raw input data is examined for suitable features before being transformed (e.g., aggregations of one or more raw input data fields) into training data [Baylor et al., 2017], [Breck et al., 2019], [Domingos, 2012], [Lin and Kohcz, 2012], [Polyzotis et al., 2018], [Rajaram et al., 2020]. Next, the data is cleaned by filtering out uncorrelated data [Cuzzocrea et al., 2011], [Polyzotis et al., 2018], specifying quality rules, detecting errors, inconsistencies and anomalies [Baylor et al., 2017], [Chu et al., 2016], [Khayyat et al., 2015], and fixing these errors [Chu et al., 2016], [Khayyat et al., 2015], [Polyzotis et al., 2018], [Volkovs et al., 2014].

To guarantee a successful preparation and cleaning of the data, each batch of data needs to be validated based on its properties [Baylor et al., 2017], [Breck et al., 2019], [Polyzotis et al., 2018], [Rajaram et al., 2020], [Schelter et al., 2018], [Volkovs et al., 2014] and potential dependencies [Polyzotis et al., 2018], deviations [Breck et al., 2019], [Polyzotis et al., 2018], or impact of features on model accuracy or performance [Domingos, 2012], [Polyzotis et al., 2018] need to be identified.

Once a model is trained, the goal of data evaluation is to evaluate the choice and encoding of the data based on the results produced by a model trained on the data, for instance by performing sanity checks [Domingos, 2012], [Polyzotis et al., 2018]. After a suitable solution was found, the newly emerging input data needs to be transformed to so-called serving data which is processible by the model [Baylor et al., 2017], [Polyzotis et al., 2018]. This usually involves the same transformation steps as required for the training data. After the serving data was successfully processed by the model, it is channeled back as training data for future iterations [Polyzotis et al., 2018].

In order to execute the aforementioned steps in an iterative and continuous manner, automated ETL tasks need to be set up. This involves the extraction of data from its

<table>
<thead>
<tr>
<th>Category</th>
<th>Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU</td>
<td>set expectations of data; How to know something (e.g. a distribution) is “right”? [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>DU</td>
<td>Analysis of features in conjunction [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>DU</td>
<td>Understanding if data reflects reality [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>DU</td>
<td>Identification of sources of data errors [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>DC</td>
<td>Dealing with data inconsistency, missing features, unit changes, ... [Polyzotis et al., 2018], [Schelter et al., 2018], [Volkovs et al., 2014]</td>
</tr>
<tr>
<td>DC &amp; DV</td>
<td>Dealing with dynamic data environments [constantly changing constraints] [Cuzzocrea et al., 2011], [Schelter et al., 2018], [Volkovs et al., 2014]</td>
</tr>
<tr>
<td>DV</td>
<td>Formulation of understandable and actionable alerts [Chu et al., 2016], [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>DP</td>
<td>Engineering set of features most predictive of the label [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>DP</td>
<td>Unused data due to data overload / too much data to be processed [Domingos, 2012]</td>
</tr>
<tr>
<td>DP</td>
<td>Feature experiments (e.g. different combinations of input features to examine their predictive value) affect multiple stakeholders (e.g. software or site reliability engineers responsible for pipeline) [Polyzotis et al., 2018]</td>
</tr>
<tr>
<td>DP &amp; DC</td>
<td>Merging data from multiple sources &amp; deal with unstructured data [Chu et al., 2016], [Cuzzocrea et al., 2011], [Dayal et al., 2009], [Hernández and Stolfo, 1998], [Schelter et al., 2018]</td>
</tr>
<tr>
<td>DP, DC &amp; DV</td>
<td>Achieving scalability of data processing and error detection in distributed settings [Chu et al., 2016], [Cuzzocrea et al., 2011], [Khayyat et al., 2015]</td>
</tr>
</tbody>
</table>

Data understanding = DU, data cleaning = DC, data validation = DV, data preparation = DP

Table 8.2: Common challenges in data management and processing
source, transporting it to a processing pipeline, transforming it to target values, and finally making it accessible to and loadable by respective machine learning models [Dayal et al., 2009], [Vassiliadis, 2009], [Vassiliadis and Simitis, 2009].

Table 8.1 provides a detailed overview of common activities in data management and data processing grouped into six categories.

As a natural consequence, these activities also entail a couple of challenges which are presented in Table 8.2 and mostly related to 1) data understanding; 2) data preparation; 3) data cleaning; and 4) data validation.

8.4.2 Model Building

In the model building phase, one or multiple models are prepared, built, and evaluated based on the previously generated input features. This is typically an iterative process that involves running an analysis, evaluating the results, and adapting or optimizing parameters and input features until an adequate solution is found [Domingos, 2012], [Lwakatare et al., 2020], [Vartak et al., 2016]. Table 8.3 outlines common activities that are part of this process.

Initially and based on the respective problem to solve, appropriate analysis techniques and model types need to be selected as part of the model preparation [Domingos, 2012], [Jordan and Mitchell, 2015], [Lin and Kolcz, 2012], [Rajaram et al., 2020]. For instance, if the input data is labeled and the goal is to classify data according to these labels, a supervised ML technique (e.g. logistic regression, support vector machines etc.) can help to achieve this. On the other side, if the requirement is to group unlabeled objects by their similarity, an unsupervised approach (e.g. k-means clustering, hierarchical clustering etc.) is the better choice.

Oftentimes, it is not advisable to use all available features as input features for the selected model as this can create noise and cause a decrease in model accuracy [Domingos, 2012]. This results in the need for feature selection techniques that aim at identifying the most relevant input features for a given model [Baylor et al., 2017], [Dash and Liu, 1997], [Lin and Kolcz, 2012].

<table>
<thead>
<tr>
<th>Activity</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model preparation</strong></td>
<td></td>
</tr>
<tr>
<td>Selection of appropriate analysis/model type</td>
<td>[Domingos, 2012], [Jordan and Mitchell, 2015], [Lin and Kolcz, 2012], [Rajaram et al., 2020]</td>
</tr>
<tr>
<td>Selection of input features</td>
<td>[Baylor et al., 2017], [Dash and Liu, 1997], [Domingos, 2012], [Lin and Kolcz, 2012]</td>
</tr>
<tr>
<td><strong>Model building</strong></td>
<td></td>
</tr>
<tr>
<td>Splitting input data into training and test set</td>
<td>[Domingos, 2012], [Lin and Kolcz, 2012]</td>
</tr>
<tr>
<td>Model training on training data</td>
<td>[Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Obston et al., 2017], [Rajaram et al., 2020], [Sparks et al., 2017], [Vartak et al., 2016]</td>
</tr>
<tr>
<td>Application of model to test data</td>
<td>[Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Sparks et al., 2017], [Vartak et al., 2016]</td>
</tr>
<tr>
<td><strong>Model evaluation</strong></td>
<td></td>
</tr>
<tr>
<td>Quality evaluation based on test results [e.g. accuracy, precision, recall, F1-score]</td>
<td>[Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Obston et al., 2017], [Rajaram et al., 2020], [Sparks et al., 2017], [Vartak et al., 2016]</td>
</tr>
<tr>
<td>Decision: accept or rework model [e.g. by adapting input features or model parameters]</td>
<td>[Bauer and Kohavi, 1999], [Domingos, 2012], [Lin and Kolcz, 2012]</td>
</tr>
</tbody>
</table>

Table 8.3: Common activities in model preparation, building, and evaluation
Table 8.4: Common challenges in model preparation, building, and evaluation

<table>
<thead>
<tr>
<th>Category</th>
<th>Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP</td>
<td>Selecting appropriate model types for a specific problem [Amershi et al., 2019], [Domingos, 2012], [Kim et al., 2017]</td>
</tr>
<tr>
<td>MP</td>
<td>Dealing with too many (irrelevant) input features [Domingos, 2012]</td>
</tr>
<tr>
<td>MP</td>
<td>Coordination and communication of involved stakeholders (e.g. ML specialists, software engineers, ...) [Amershi et al., 2019], [Arpteg et al., 2018], [Kim et al., 2017]</td>
</tr>
<tr>
<td>MB</td>
<td>Avoidance of overfitting [Domingos, 2012]</td>
</tr>
<tr>
<td>MB</td>
<td>Debugging of ML models [Amershi et al., 2019], [Arpteg et al., 2018], [Tata et al., 2017]</td>
</tr>
<tr>
<td>ME</td>
<td>Defining quality specifications (e.g. “when is the prediction quality good enough?”, “is the model save to serve?”) [Baylor et al., 2017]</td>
</tr>
</tbody>
</table>

Model preparation = MP, model building = MB, model evaluation = ME

and Liu, 1997], [Domingos, 2012], [Lin and Kolcz, 2012]. Once the input data is filtered according to the determined feature relevance, a training and a test data set need to be created as part of the model building phase [Domingos, 2012], [Lin and Kolcz, 2012]. In a next step, the training set is used to train a model that was selected to solve a specific problem [Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Rajaram et al., 2020], [Olston et al., 2017], [Sparks et al., 2017], [Vartak et al., 2016]. Before being able to validate the quality of the model, it is applied on the test set to investigate how it performs on previously unseen data [Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Sparks et al., 2017], [Vartak et al., 2016]. Consequently, the results of this step can be used for the model evaluation. There are several metrics that support practitioners in assessing the quality of their models, for instance by calculating the accuracy, precision, recall, or F1-score [Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Olston et al., 2017], [Rajaram et al., 2020], [Sparks et al., 2017], [Vartak et al., 2016]. Based on the evaluation results, the model can either be accepted as a reasonable solution or it needs to be reworked, for example by adapting the parameters or input features that are used for training the model [Bauer and Kohavi, 1999], [Domingos, 2012], [Lin and Kolcz, 2012]. This iterative process is often accompanied by several challenges. Table 8.4 summarizes a few of these challenges that we feel like are most important for our study. The presented challenges are categorized in model preparation, model building, and model evaluation.

8.4.3 Model Deployment and Serving

In order to fully leverage the benefits of ML to gain valuable insights, it is crucial to go beyond prototypical analyses by deploying models in production where they are actually used [Lin and Kolcz, 2012]. It has even been observed that “organizations that make the most of machine learning are those that have in place an infrastructure that makes experimenting with many different learners, data sources, and learning problems easy and efficient” [Domingos, 2012]. For that reason, we summarize the common activities in model deployment and model serving in Table 8.5. Deployment infrastructures for ML models often consist of multiple components each
<table>
<thead>
<tr>
<th>Activity</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
</tr>
<tr>
<td>Component for model validation (before serving &amp; often coupled with data validation)</td>
<td>[Baylor et al., 2017], [Crankshaw et al., 2015], [Olston et al., 2017]</td>
</tr>
<tr>
<td>Component for continuous model evaluation &amp; monitoring (performance, quality,...)</td>
<td>[Baylor et al., 2017], [Crankshaw et al., 2015], [Rajaram et al., 2020], [Olston et al., 2017]</td>
</tr>
<tr>
<td>Component or serving solution to deploy model in production</td>
<td>[Baylor et al., 2017], [Crankshaw et al., 2015], [Lin and Kolcz, 2012]</td>
</tr>
<tr>
<td>Component for monitoring pipelines (checkpoint after each pipeline)</td>
<td>[Rajaram et al., 2020], [Sparks et al., 2017]</td>
</tr>
<tr>
<td><strong>Setups</strong></td>
<td></td>
</tr>
<tr>
<td>Setup model lifecycle management (to keep overview of deployed models)</td>
<td>[Crankshaw et al., 2015], [Vartak et al., 2016]</td>
</tr>
<tr>
<td>Setup workflow manager for job coordination</td>
<td>[Crankshaw et al., 2015], [Lin and Kolcz, 2012]</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td></td>
</tr>
<tr>
<td>Loading new model before unloading old model</td>
<td>[Olston et al., 2017]</td>
</tr>
<tr>
<td>Validation of model and serving infrastructure (incl. reliability checks) before pushing to production environment</td>
<td>[Baylor et al., 2017], [Crankshaw et al., 2015], [Olston et al., 2017]</td>
</tr>
<tr>
<td>Continuous application of model to (new) serving data</td>
<td>[Crankshaw et al., 2015], [Lin and Kolcz, 2012], [Rajaram et al., 2020]</td>
</tr>
<tr>
<td>Continuous model evaluation / monitoring</td>
<td>[Baylor et al., 2017], [Crankshaw et al., 2015], [Olston et al., 2017], [Rajaram et al., 2020]</td>
</tr>
<tr>
<td>Rollback in case of errors</td>
<td>[Olston et al., 2017], [Rajaram et al., 2020]</td>
</tr>
<tr>
<td>Periodically update models</td>
<td>[Crankshaw et al., 2015], [Lin and Kolcz, 2012]</td>
</tr>
</tbody>
</table>

Table 8.5: Common activities in model deployment and model serving

For a specific task and executable as an automated workflow coordinated by a workflow manager. Besides the component that handles the actual deployment of models in production [Baylor et al., 2017], [Crankshaw et al., 2015], [Lin and Kolcz, 2012], it is advisable to have additional components for validating the model before deployment [Baylor et al., 2017], [Crankshaw et al., 2015], [Olston et al., 2017], for continuously evaluating and monitoring the model after being deployed in production [Baylor et al., 2017], [Crankshaw et al., 2015], [Rajaram et al., 2020], [Olston et al., 2017], and for monitoring if all pipelines are up and running as expected [Rajaram et al., 2020], [Sparks et al., 2017].

In addition to the components, a few setups are required for automating the deployments while keeping an overview of the deployed models. For one, a model lifecycle management should be set up that allows the comparison and monitoring of models over time and provides information on the currently deployed models [Crankshaw et al., 2015], [Vartak et al., 2016]. For another, the jobs required to deploy a model can be coordinated an executed using a workflow manager [Crankshaw et al., 2015], [Lin and Kolcz, 2012]. After triggering the workflow, the model is updated in an automated manner and in case of errors a predefined rollback plan is executed.

In general, the process of model deployment and model serving requires the following steps which are typically encapsulated in respective components: First, a new model is loaded for deployment before unloading the old model [Olston et al., 2017]. In a next step, the model as well as the serving infrastructure are validated (e.g. reliability checks) before pushing the new model to the production environment [Baylor et al., 2017], [Crankshaw et al., 2015] [Olston et al., 2017].

Once the model is deployed to production, it can be used and continuously applied to newly emerging serving data [Crankshaw et al., 2015], [Lin and Kolcz, 2012], [Rajaram
et al., 2020]. In order to guarantee that the model works as expected, a continuous evaluation and monitoring of the model is required [Baylor et al., 2017], [Crankshaw et al., 2015], [Olston et al., 2017], [Rajaram et al., 2020]. In case the model does not behave as expected, a rollback plan is executed and typically the current model is replaced by a previous well-working version of the model [Olston et al., 2017], [Rajaram et al., 2020]. Following this process, models can be periodically updated and deployed to production [Crankshaw et al., 2015], [Lin and Kolcz, 2012].

Analogously to data management and model building, model deployment and model serving also entails several challenges. Four of the key challenges are presented in Table 8.6. The challenges are categorized into infrastructure and model-specific topics.

<table>
<thead>
<tr>
<th>Category</th>
<th>Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Integration of third-party packages or tools [Lin and Kolcz, 2012], [Sculley et al., 2015]</td>
</tr>
<tr>
<td>I</td>
<td>Brittle pipelines / &quot;pipeline jungle&quot; [Lin and Kolcz, 2012], [Sculley et al., 2015]</td>
</tr>
<tr>
<td>M</td>
<td>Managing and monitoring multiple models [Crankshaw et al., 2015], [Sculley et al., 2015], [Vartak et al., 2016]</td>
</tr>
<tr>
<td>M</td>
<td>Dealing with expected and unexpected variations during model evaluation [Baylor et al., 2017], [Sculley et al., 2015]</td>
</tr>
</tbody>
</table>

Table 8.6: Common challenges in model deployment and model serving

### 8.5 Framework Derivation

Based on the insights gained from the literature review, we derive a framework for supporting an end-to-end development and deployment of ML models in the context of software analytics and business intelligence (see Figure 8-2).

While the literature review examines the topics data management and processing, model building, and model deployment and serving individually, in reality a separation of the three is not that trivial. In fact, for building end-to-end solutions the fields are very much interrelated as the activities depend on each other and sometimes even overlap.

Oftentimes, ML projects start out as a prototypical analysis due to a limited amount of time and resources [Figalist et al., 2020], [Sculley et al., 2015]. In order to use and actually benefit from the ML model, it needs to be deployed to a production environment which can be time and cost-intensive but nonetheless crucial [Lin and Kolcz, 2012], [Sculley et al., 2015]. To avoid the deployed models from being outdated, it is important to provide a functionality for dynamically deploying new models or iteratively retraining and updating existing models [Crankshaw et al., 2015], [Lin and Kolcz, 2012].

As a result, we identify three iterative cycles which are passed through during an end-to-end development of ML solutions and, therefore, serve as the main dimensions in our framework: 1) Prototyping cycle (blue), 2) deployment cycle (green), and 3) update cycle (orange).
8.5.1 Prototyping Cycle

In software analytics and business intelligence, relevant input data typically emerges from multiple sources that need to be extracted, set in relation, and stored in a common data storage [Figaliste et al., 2020]. As part of the data preparation, potential input features can be identified and extracted based on a snapshot of raw data [Domingos, 2012], [Lin and Kolcz, 2012], [Polyzotis et al., 2018]. To ensure a sufficient data quality, a couple of data cleaning activities need to be performed (e.g., filtering, consistency checks, or error detection techniques and repairs) [Baylor et al., 2017], [Chu et al., 2016], [Khayyat et al., 2015], [Polyzotis et al., 2018], [Volkovs et al., 2014]. Depending on the overarching goal of the analysis, appropriate ML models have to be selected that are suitable to achieve a specific task [Domingos, 2012], [Jordan and Mitchell, 2015], [Lin and Kolcz, 2012], [Rajaram et al., 2020]. Based on the selected model, it is recommended to apply feature selection techniques to the input data set to identify a subset of the most relevant input features [Baylor et al., 2017], [Dash and Liu, 1997], [Domingos, 2012], [Lin and Kolcz, 2012]. This subset is then extracted from the overall data set and the values of each input features are standardized. Before training the model, the respective subset should be split into a training and a test data set (usually 70%/30% or 80%/20%) [Domingos, 2012], [Lin and Kolcz, 2012].
Training the model using training data and testing it on test data, allows an examination of how well the model performs on previously unseen data [Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Sparks et al., 2017], [Vartak et al., 2016]. Based on the test results, the choice of model parameters and input features should be evaluated [Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Olston et al., 2017], [Rajaram et al., 2020], [Sparks et al., 2017], [Vartak et al., 2016]. If the evaluation indicates a decent quality of the model (e.g. based on accuracy, precision, recall, and F1-score), it can be cleared for deployment. Otherwise, the cycle is run through again and the model is reworked until its quality reaches a desired level.

8.5.2 Deployment Cycle

Taking a ML model to production involves much more than only model deployment. For one, a ETL job needs to be set up that continuously extracts, transforms, and loads the latest input and serving data [Dayal et al., 2009], [Vassiliadis, 2009], [Vassiliadis and Simitsis, 2009]. Before a new model is deployed, it is loaded and validated to avoid errors of faulty behavior in the production environment [Baylor et al., 2017], [Crankshaw et al., 2015], [Olston et al., 2017].

For each batch of new data a data validation pipeline is triggered that checks the data for anomalies [Breck et al., 2019], [Schelter et al., 2018], [Volkovs et al., 2014]. After one or more iterations, the update cycle can be entered at this point in case the model needs to be retrained which is evaluated during the model evaluation later on. Analogous to the training data, the new input data is transformed to serving data [Baylor et al., 2017], [Polyzotis et al., 2018]. This involves data cleaning, feature extraction and standardization. In a next step, the model can be applied to the new and preprocessed data [Crankshaw et al., 2015], [Lin and Kolcz, 2012], [Rajaram et al., 2020].

Since both data and model behavior evolves over time, it is crucial to continuously evaluate the model performance and its input data [Baylor et al., 2017], [Crankshaw et al., 2015], [Olston et al., 2017], [Rajaram et al., 2020]. If the model does not perform as expected, a retraining of the model is triggered for the next iteration [Crankshaw et al., 2015], [Lin and Kolcz, 2012].

In addition to this, the results of the analysis need to be visualized. By explaining the results as intuitively as possible, users of SA/BI solution will be able to understand and interpret the results and turn it into actionable insights [Figalist et al., 2020]. In the last step before the cycle is repeated from the beginning, the recently processed serving data is channeled back as training data which will be included in upcoming retrainings of the model [Polyzotis et al., 2018].

Since these steps are typically automated in one or multiple pipelines, it is important to implement several checkpoints along the way, that continuously monitor and check whether each task is working properly [Rajaram et al., 2020], [Sparks et al., 2017]. In case of errors or anomalies, an alert should be sent to the respective stakeholder.
8.5.3 Update Cycle

As a natural consequence of constantly evolving data, the model’s accuracy can start to decrease some time after being deployed [Lin and Kolcz, 2012]. As soon as this is detected during the model evaluation in the deployment cycle, a retraining of the model will be triggered after the upcoming data validation. An updated data set is created that consists of the initial training data as well as the new serving data that was channelled back as training data [Polyzotis et al., 2018]. Analogous to the initial training only the relevant input features are extracted, standardized and split into a training and a test set. The model is retrained based on the new training set and tested on the test set respectively [Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Sparks et al., 2017], [Vartak et al., 2016]. Based on the results, the model’s input features and parameters needs to be evaluated before deciding whether to improve the model’s quality in an additional iteration or whether to clear it for deployment and add both the current and the new model to the model management [Baylor et al., 2017], [Domingos, 2012], [Lin and Kolcz, 2012], [Olston et al., 2017], [Rajaram et al., 2020], [Sparks et al., 2017], [Vartak et al., 2016]. The latter enables a clear overview of all models and allows an easy rollback in case of erroneous behavior in production.

8.6 Framework Validation

In order to validate the applicability of our framework in practice, we use a real-world ML-based SA/BI solution that is currently being developed for an industrial platform provider to 1) compare the activities of the framework to the actual activities executed in practice; and 2) to strategically plan and direct upcoming activities to finalize the end-to-end implementation.

8.6.1 Current Status

At the beginning of our collaboration, the product managers were interested in running customized analyses on their customers’ usage data. Specifically, whenever a customer’s action triggers a request to the platform or one of its applications, it is tracked in the platform and app usage logs. The platform itself is based on AWS. Therefore, we decided to setup the custom ML-based SA/BI solution using the existing AWS infrastructure and services.

Currently, the platform and app usage logs produce 100GB of data every day. For this reason, the data is aggregated and stored in a compressed format (26GB per day) in AWS S3 buckets\(^2\). The log data is available for the past 1.5 years and, in addition to that, we also have access to the sales data that keeps track of which customer purchased what kind of licenses.

\(^2\text{https://aws.amazon.com/s3/}\)
Prototyping Cycle

As the future users of the system, the product managers were interested in analyzing customer churn for the applications hosted on the platform as a first use case. In the beginning, we focus on one specific application to build a first prototype. Therefore, we identify potential input features based on the information that is available in the logs (e.g. user id, http status code, relative URL path) and pre-filter the data by the selected application. We setup a script that extracts and aggregates the input features on customer level ($n=174$) while enriching and labeling it with the sales data (binary label for churn/non-churn).

Next, we applied several different supervised ML models (support vector machines, decision trees, logistic regression, neural network) to the data set in an iterative manner. We apply principal component analysis to the standardized input data in order to identify the most relevant subset of features, before splitting the data set into a training and a test set.

Based on this, the models were trained on the training data and tested using the test set. It took us several iterations and experiments with different models, model parameters and input features before ending up with the final model that is being deployed in the upcoming step.

Deployment Cycle

At the current state of our ML-based SA/BI solution, we have not yet completed the deployment cycle. Before being able to deploy the model, we had to come up with an efficient, reliable and robust solution for handling the enormous amounts of data (100GB per day). It was a complex task to get an overview of the data, to define which fields to keep for long-term storage, and finally to specify the format to store it in.

After coming up with a concept for this, one of the software architects setup pipelines that continuously extract, transform, compress and load the latest log data into a S3 bucket using the specified format. In addition to that, he also setup data validation pipelines that check each batch of new data for anomalies and inconsistencies. It is important to continuously monitor all pipelines. During one of the interviews, the software architect explains that “we have to make sure the pipelines are not failing for whatever reasons and if they’re failing we’re notified and can restart them”. Moreover, they need to ensure that “the pipeline elements that are doing the preprocessing are always up and triggered at appropriate times". The software architect also notes that it "requires a lot of engineering effort to keep the pipeline running in a correct manner."

After the data pipelines are set up, we load and deploy our ML model using Amazon SageMaker$^3$. The SageMaker modules for Python offer out-of-the-box functionalities for deploying ML models to an AWS instance that are accessible via an API (see Figure 8-3).

$^3$https://aws.amazon.com/sagemaker/
8.6.2 Planning and Evolution

In order to use the deployed model to make actual predictions, we now plan and execute the remaining steps following the presented framework.

Deployment Cycle

In the upcoming step, the model is applied to new serving data. This constitutes a bit of a challenge as up until now all extracted data is stored in the same S3 bucket. As a result, we now need to create an additional S3 bucket for storing the serving data. In order to preprocess the newly emerged data to serving data, we can reuse the script created during prototyping for transforming and extracting the input features out of the raw data.

In order to continuously evaluate the model in production, we setup Amazon SageMaker’s Model Monitor that provides summary statistics, detects concept drifts and indicates when a model needs to be retrained. In order to perform potential retrainings on the newest data available, we transfer the data from the serving S3 bucket to the training S3 bucket once it was processed by the model. Lastly, we plan to visualize the results in Amazon QuickSight\(^4\).

Update Cycle

For continuous updates of models, AWS offers the Step Functions Data Science SDK\(^5\) for Amazon Sagemaker to automate model retraining and deployment. An ETL job is setup to extract and preprocess the latest data. Following this, a new model is trained and evaluated. If the model accuracy is above a certain threshold (e.g. 90%), a new endpoint is created for deployment and the model is added to the model management.

---

\(^4\)https://aws.amazon.com/quicksight/

\(^5\) https://docs.aws.amazon.com/step-functions/latest/dg/concepts-python-sdk.html
Findings - planning and evolution: the framework supported us in keeping an overview of remaining tasks; by following the cycles and activities in the framework the definition of the roadmap and next steps was efficient and easy; we were able to quickly identify errors or missing components in our original approach (storage of training and serving data)

8.7 Conclusion

Gaining customized insights on product or usage behavior can be a valuable asset for many stakeholders involved in software-intensive businesses which results in a need for ML-based SA/BI solutions. Building and, more importantly, deploying and maintaining such solutions is, however, time-consuming, complex and burdensome as it requires knowledge from several different domains.

For this reason, we scanned existing literature on data management and processing, model building, and model deployment to derive a framework that comprises all key activities from data collection to retraining of deployed models. In addition to that, our framework is structured in three iterative cycles: a prototyping cycle, deployment cycle, and an update cycle. These cycles resemble stages in the lifecycle of a ML model and by outlining the transitions between stages, our framework specifically guides the journey from a prototypical analysis to a productively running ML model. The results of the validation indicate that the activities of the framework are consistent with the activities performed in practice. Moreover, the framework is a practical tool to keep an overview of all required steps and to efficiently define and plan upcoming activities. Moreover, we observed that the separation of activities across the conceptual phases creates the perception that the overall, potentially overwhelming process now consists of several smaller ones that are easier to handle.

One limitations of our study is the development state of our ML-based SA/BI solution. As we are still in the process of implementing parts of the deployment and update cycle, we are only partially able to compare the framework’s activities to the activities executed in practice. Further research could, therefore, be dedicated to a long-term validation of the framework based on already established SA/BI solutions and to identifying remaining challenges and needs for more in-depth guidance by practitioners to adapt the framework to their needs.
Chapter 9

A Multi-Case Study on Driving the Adoption of a DataOps Mindset in B2B Software-Intensive Companies


Abstract

Nowadays, many software-intensive companies collect operational data to draw conclusions and gain insights for continuous improvement. To collect, manage, process, and analyze data from various sources, the concept of DataOps emerged as an extension of DevOps for data analytics. It enables organizations to shorten their data analysis lifecycles and to continuously improve their product by making decisions based on data. However, the adoption of DataOps within organizations is difficult as it requires a major change of mindset. All team members need to have confidence in and understanding of the data to change their decision making processes. We conduct an exploratory case study of four product teams to investigate the adoption of a DataOps mindset in B2B software-intensive companies. The results show that it is crucial to adapt different steps in the adoption process to the stakeholders’ needs and mindsets. We identify four stakeholder phases that require different types of interaction between stakeholders and data engineers and data scientists. Additionally, we derive a model combining data-driven practices applicable to B2B-specific data characteristics with respective stakeholder interactions to incrementally establish a DataOps mindset. The stakeholder phases in combination with the model enable practitioners to tailor the adoption process to their team’s needs.
9.1 Introduction

In order to speed up delivery time and shorten release cycles, many software-intensive companies evolved and shifted from agile software development to DevOps [Ebert et al., 2016] [Leite et al., 2019]. DevOps aims at bringing previously separated teams, especially development and operations, closer together to develop and deploy resilient and high-quality software [Dyck et al., 2015] [Erich et al., 2017].

Domain-specific characteristics and needs led to various adaptations and extensions of existing DevOps principles [Capizzi et al., 2019], e.g. in security (SecDevOps) [Mohan and Othmane, 2016], web [Sacks, 2012], artificial intelligence (AIOps) [Dang et al., 2019] and machine learning (MLOps) [Karamitsos et al., 2020], and data management (DataOps) [Ereth, 2018]. Taking into consideration that monitoring infrastructure and system or end-user behavior is an essential part of DevOps leading to large amounts of data to be handled and processed, the need for DataOps becomes more and more present [Capizzi et al., 2019]. DataOps can be defined as "a set of practices, processes and technologies that combines an integrated and process-oriented perspective on data with automation and methods from agile software engineering to improve quality, speed, and collaboration and promote a culture of continuous improvement" [Ereth, 2018]. While it can be seen as an enabler for data analytics and machine learning, the adoption of DataOps entails the need for organizational change and a shift of mindset [Ereth, 2018]. Decisions that were previously made on the basis of opinions and gut feelings are now based on data in order to enable faster feedback cycles [Fabijan et al., 2018] [Olsson and Bosch, 2019] but also requires a large amount of trust and believe in the value of working in a data-driven way [Ereth, 2018] [Thussoo, 2017]. This often impedes the adoption of data-driven ways of working [Figalist et al., 2020].

During the adoption of DataOps, data scientists and data engineers are becoming part of traditional software engineering teams [Kim et al., 2016] [Kim et al., 2017]. However, convincing stakeholders of the value and benefits of working in a data-driven way, is proven to be very challenging [Kim et al., 2017] [Figalist et al., 2020]. Data availability and data quality is a well-known issue for data scientists in software engineering teams [Kim et al., 2017]. In business-to-business (B2B) contexts it is particularly challenging since the amount of available data is often significantly lower as compared to business-to-customer (B2C) domains [Bohanec et al., 2017]. For one, this is caused by a more limited number of customers in B2B contexts which is a typical characteristic of B2B businesses [Russo et al., 2016]. For another, the customer of a software-intensive product is often not the end-user of the product but rather an intermediate entity [Figalist et al., 2019b]. As a result, data generated by the end-user is often more complex to analyze or not accessible at all. These B2B specific characteristics make it difficult to work with data in a continuous way. Ultimately, the required shift of mindset and organizational changes, the lack of trust and believe in its value, and the challenges of dealing with data in a continuous way impede the adoption of DataOps practices and data-driven approaches that are building on it. To this point, the amount of research on the adoption of DataOps is quite scarce. While recent literature covers the adoption from a technical point of view [Munappy
et al., 2020], the process of establishing a DataOps mindset within teams has not yet been investigated in great detail.

In order to address this, we conducted an exploratory case study within four product teams across three companies that are developing and operating B2B online platforms and applications in various domains. The product teams are either at the beginning or in the middle of transitioning from DevOps to DataOps. We supported the product teams as data engineers and data scientists and work with multiple non-data-science stakeholders of each product team to assist them during the incremental adoption of DataOps and data-driven practices. During the course of the case study, we found that it is crucial to adapt different steps in the adoption process to the stakeholders’ needs and mindsets. We were able to identify four stakeholder phases that participants in our study went through. For each of the phases, we describe how the interaction with stakeholders can be adapted in order to meet their individual needs. Moreover, we summarize the individual steps taken across the four cases. Based on our observations, we derive a generic model for driving the adoption of a DataOps mindset in B2B software-intensive companies which is instantiable in each of the identified phases.

The contribution of our study is two-fold: For one, we introduce the four stakeholder phases that the participants of our case study went through. Each phase requires a different type of interaction between stakeholders and data engineers and data scientists. Depending on the phase, practitioners who are pushing towards the adoption of DataOps can adapt their interaction with other stakeholders accordingly. For another, we present a model combining data-driven practices applicable to B2B specific data characteristics with respective stakeholder interactions in order to incrementally establish a DataOps mindset. In addition to that, the model is adaptable to the four stakeholder phases in order to capture the individual needs in each of different stakeholders in all stages of the adoption process. By better understanding how to work and interact with stakeholders in different phases, practitioners can use the model in combination with the stakeholder phases to incrementally drive the adoption of a DataOps mindset within their respective teams.

The remainder of the paper is structured as follows: Section 9.2 provides the background of our study while Section 9.3 outlines the research method and study design. The case study is described in Section 9.4 followed by the findings in Section 9.5. Section 9.6 gives an overview of the related work while the threats to validity are outlined in Section 9.7 before concluding in Section 9.8.

9.2 Background

9.2.1 Agile and DevOps

Nowadays, the use of agile methods in software development is widespread. According to recent estimations, the majority of software-intensive companies are using agile methodologies for developing their software [Hemon et al., 2020]. It allows software development teams to react timely to changing customer needs and requirements [Co-
hen et al., 2004] and to shorten their release cycles [Hemon et al., 2020]. While this holds true for the development activities, operations who take care of the actual software release often appeared to be a bottleneck [Hemon et al., 2020]. Consequently, the concept of DevOps originated in order to address this problem and the number of companies moving towards DevOps is increasing ever since [Hemon et al., 2020].

"DevOps is a development methodology aimed at bridging the gap between Development (Dev) and Operations (Ops), emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices" [Jabbari et al., 2016]. It aims at bringing previously separated teams, especially development and operations, closer together in order to develop and deploy resilient and high-quality software in short release cycles [Dyck et al., 2015]. A shift towards DevOps involves both technical but also cultural transformations [Erich et al., 2017] [Leite et al., 2019] [Lwakatare et al., 2016] [Zhu et al., 2016]. Technical transformations include, for instance, automated deployments using build and continuous integration tools, treating infrastructure as code, and continuous monitoring of infrastructure and system behavior in production [Ebert et al., 2016] [Erich et al., 2017] [Lwakatare et al., 2016]. On the organizational side, it is crucial to build and strengthen a collaborative culture based on trust to successfully establish a straightforward communication and shared responsibilities within and across teams [Luz et al., 2018] [Walls, 2013].

9.2.2 DataOps and Data-Driven Software Engineering

With the possibility to collect, store, and analyze data at every step of the DevOps lifecycle in order to enable short feedback cycles, there is a rising need for adopting DataOps practices in order to handle data management and make effective use of the data. Analogous to the transformation towards DevOps, a transition towards DataOps does not only entail changes and new ways of working on a technical but also on a cultural and organizational level [Ereth, 2018]. However, since organizational transformations entail a change of mindset by the individuals within an organization and across hierarchies, it is considered to be a highly challenging task. The transformation towards data-driven ways of working is often treated with rejection and resistant behavior since the data does not always match with everyone’s believes and expectations which leads to a lack of acceptance. A previous study showed that people often try to find explanations whenever the data conflicts with their beliefs in order to “make their beliefs still true" [Olsson and Bosch, 2019]. However, it is also important not to trust data blindly, but to understand its explanatory power and its limitations. Changing mindsets and establishing a new culture within an organization is a continuous process that relies on trust, enthusiasm, and confidence in the new way of working [Walls, 2013]. Therefore, it is important to empower non-data-science stakeholders to work with data and to understand and interpret it in the right way [Ereth, 2018]. It is essential to incrementally build up this trust in order to clearly highlight the need for and benefits of an organizational transformation [Walls, 2013]. Collecting and making use of data to improve software-intensive products or processes in a data-driven way, has become a well-known concept, e.g. as part of software an-
alytics [Buse and Zimmermann, 2012] [Menzies and Zimmermann, 2013], business intelligence [Negash and Gray, 2008], or online controlled experiments [Kohavi et al., 2013] [Kohavi and Longbotham, 2017]. To achieve this in a continuous way, DataOps practices play an important role [Ereth, 2018]. In order to establish data-driven decision making, data-analytical thinking is “important not just for the data scientist but throughout the organization” [Provost and Fawcett, 2013]. Since decisions are no longer based on opinions and gut feelings but on actual data, fundamental changes in how people are used to work are required for a successful adoption. To establish this, companies like Microsoft foster a close collaboration between data scientists and software development teams [Kim et al., 2016] [Kim et al., 2017]. However, bringing data scientists and data engineers into traditional software engineering teams also entails a risk for emerging conflicts. For one, it is challenging to convince teams of the value generated by data science and, for another, it is difficult to “convey the resulting insights to leaders and stakeholders in an effective manner” [Kim et al., 2017]. It is important that all team members promote a close and continuous collaboration. This includes jointly interpreting available data, identifying and refining questions, and explaining the results in an understandable and interpretable way [Kim et al., 2016].

Moreover, the quality of an analysis, which is enabled by DataOps, and the quality and usefulness of its corresponding results highly depend on the availability and quality of data [Gudivada et al., 2017] [Polyzotis et al., 2018]. Precisely this often causes uncertainty in applying data-driven approaches in business-to-business (B2B) contexts. Oftentimes, the volume of data being generated or even available in B2B contexts is significantly lower as compared to business-to-customer (B2C) domains [Bohanec et al., 2017]. For one, this is caused by a more limited number of customers in B2B contexts which is a typical characteristic of B2B businesses [Russo et al., 2016]. For another, the customer of a software-intensive product is often not the end-user of the product but rather an intermediate entity [Figalisi et al., 2019b]. As a result, data generated by the end-user is often more complex to analyze or not accessible at all. These limitations often lead to a lack of confidence in the data which in turn makes it difficult to drive organizational and cultural changes [Walls, 2013]. To this date, the amount of research on the adoption of DataOps is very limited. So far, existing studies mostly focus on the adoption from a technical point of view [Munappay et al., 2020] while the process of establishing a required DataOps mindset within teams has not yet been examined in great detail. To address this, our study aims at investigating how to drive the adoption of a DataOps mindset in the context of B2B software-intensive companies.

9.3 Research Method & Study Design

We selected an inductive research approach [Goddard and Melville, 2004] for our study in order to derive the model presented in this paper from multiple, real-world cases. Specifically, we conducted an exploratory case study which was designed by following the guidelines from Runeson and Höst [Runeson and Höst, 2009]. The
objective of this study was the investigation of how the adoption of DataOps and data-driven practices can be promoted in B2B software-intensive companies. Therefore, the studied case is the adoption of these practices within multiple, real-world B2B product teams. We aim at addressing the following research question: *How to drive the adoption of a DataOps mindset in B2B software intensive companies?*

In order to identify cases for our study, we sent out a description of the study and an invitation to participate to contact persons of various platforms and applications who shared it among their teams. We received a positive response from four different platform or application providers of three companies established in a variety of domains in B2B markets. In total, eleven stakeholders with different roles (product management, operations engineers, software architects, sales, CTO) across the four platforms and application participated in the study.

### 9.3.1 Case Platforms and Study Participants

Table 9.1 gives an overview of the cases and the corresponding companies and stakeholders. In addition to that, the following subsections provide a more detailed description of the platforms and application as well as the participating stakeholders.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Case Company &amp; Product</th>
<th>Stakeholders</th>
</tr>
</thead>
</table>
| **Case A** | Healthcare platform (Company A) | • Product owner  
• Operations manager  
• Two operations engineers |
| **Case B** | Industrial platform (Company B) | • Product manager  
• Software architect  
• Two sales representatives |
| **Case C** | Application hosted on industrial platform (Company B) | • Product manager |
| **Case D** | Advertising analytics platform (Company C) | • CTO  
• VP of product |

Table 9.1: Overview of case products and stakeholders participating in the study

**Healthcare Platform**

*Case A* was conducted in *Company A* which is a subsidiary company of *Company B*. More precisely, the case is based on a platform provider established in the healthcare domain. Several platform-internal and external medical applications are hosted on the platform. Over the course of this study, we worked with one product owner, one operations manager, and two operations engineers of the platform. As a team they had set themselves the goal to collect usage data of the platform as well as the applications in order to enable the development teams to make use of the data and work in a more data-driven way. For this reason, they were interested in participating in the study.
Industrial Platform & Application

The second case of our study, Case B, is based on a platform provider of Company B operating in the industrial domain. The platform hosts several company-internal as well as external applications for industrial device management and analysis. One of the applications served as Case C in our study. The application offers data visualization functionalities to its end customers for visualizing their device data. For Case B we collaborated with one product manager, one software architect, and two sales representatives of the industrial platform provider and for Case C we worked with the application’s product manager.

The product manager and software architect of the platform provider work in a team dedicated to collecting, processing and storing the platform’s and the application’s usage data logs. Based on these logs, they provide services to application providers to examine how their application is being used in a very basic way. Since they wanted to shift towards more complex analyses, they were interested in participating in our study.

Advertising Analytics Platform

Case D of our study is based on a platform provider of Company C operating in the advertising domain. The platform offers a variety of functionalities to create and monitor advertising campaigns and to create and export reports and dashboards of the campaigns. During the course of this study, we collaborated with the platform’s chief technology officer (CTO) and the vice president (VP) of product.

Both of them were highly interested in adopting a more data-driven way of working but they were also facing B2B-related data size limitations. For this reason, they were interested to participate in the study in order to find ways to analyze the data nonetheless.

9.3.2 Research Process

In the first step of the research process, the overarching research questions was defined (How to drive the adoption of a DataOps mindset in B2B software intensive companies?). In order to identify relevant literature, we reviewed existing literature in the area of DevOps and DataOps. More precisely, we queried scientific libraries (Springer Link, ScienceDirect, ACM Digital Library, IEEEXplore) using search terms related to our research question, e.g. “(DevOps OR DataOps) AND (practices OR mindset OR culture OR adoption OR transition OR transformation OR B2B)”. We did not conduct a systematic literature review but informally examined the search results with the primary goal of generating a basis for the case study and the conversations with the participants.

Following this, we scheduled introductory sessions with each of the stakeholders participating in our study. Each session was considered an unstructured interview that lasted approximately one hour and was intended to get a better understanding of the participants’ interests and pain points regarding DataOps and data-driven ways of
working and to talk about use cases to implement for the stakeholders. These insights were then used as input for working with the individual teams and for supporting the teams in data engineering and data science tasks. The collaboration and working mode was dynamically adapted to the teams’ needs and availability. However, all stakeholders were invited to (preliminary) result presentations during which they were also asked to provide feedback and additional ideas or use cases.

![Figure 9-1: Research process](image)

After working with eleven stakeholders across four cases over a period of two and a half years, we analyzed the empirical data collected during the collaboration. Specifically, we documented and compared 1) the way stakeholders expressed their information needs and the way use cases were defined; 2) the data engineering and data analysis activities; 3) how actively stakeholders wanted to be involved; and 4) the feedback, comments, and thoughts provided by stakeholders. Based on this, we derived 1) stakeholder phases that represent the collaboration with stakeholders along two dimensions: balance of initiative and the progress of adopting DataOps practices; and 2) a model for driving the adoption of a DataOps mindset in B2B software-intensive companies which is adaptable to each of the identified phases. The model in combination with the stakeholder phases highlight how to incrementally drive the adoption of a DataOps mindset by adapting the interaction with stakeholders according to their evolving needs. A step-wise summary of the research process is provided in Figure 9-1.

### 9.4 Case Study

Over the course of two and a half years, we have worked with eleven stakeholders of four platforms or products across two different companies. For each case we assessed the current status of transitioning from DevOps to DataOps. Furthermore, we collaborated with the stakeholders and actively contributed to data engineering and data analysis tasks in order to incrementally support their teams to make use of their data.

#### 9.4.1 Case A: Healthcare Platform

For Case A we collaborated with the operations team of the healthcare platform and specifically with one product owner, one operations manager, and two operations engineers (one of which also takes on data science tasks).
Initial Situation

The platform is developed and operated by modularized teams, each of them responsible for one specific component of the platform or one specific application hosted on the platform. At the time of the collaboration, the teams were in the middle of transitioning from DevOps to DataOps. They are using Azure DevOps\(^1\) and the associated services (incl. Azure Repos\(^2\) and Azure Pipelines\(^3\) for build, test and deployment with CI/CD) for the development and operation of their platform and its applications. The operations team has set itself the goal to collect performance and usage data of the entire platform and all applications in order to make it available to all other teams for further analysis. Continuous improvement based on data is an essential part of the DataOps culture [Ereth, 2018]. However, not all teams were sharing the same mindset in this regard. Some were already convinced that data could provide them with valuable insights but they do not yet know how to make actual use of it while others were more skeptical or argued that they are too busy with their daily work to adopt a new way of working.

In order to establish a common mindset, the operations team wanted to implement some first proof of concepts in order to build up trust and prove the value of working in a data-driven way. The team was already using the Azure services Application Insights\(^4\) and Log Analytics\(^5\) for monitoring the platform’s and its applications’ performance and usage. To enable long-term use of the data, they set up an infrastructure to continuously aggregate the data on a daily basis and store it in a KustoDB\(^6\). In addition to that, they merged other data sources (e.g. sales data and master data records) into the same database to simplify the use and mapping of multiple data sources. Moreover, the operations team started to create dashboards for other teams and stakeholders (e.g. sales or product management) using Microsoft Power BI on Azure\(^7\) which is directly connected to the KustoDB.

Implementation & Collaboration

Based on this initial situation, the first author joined the operations team as a data scientist. Over a period of one and a half years, the author spent one day every other week at the platform provider’s office closely collaborating with the operations team. This also included bi-weekly meetings for assessing information needs, discussing results, and brainstorming additional use cases. During the first sessions, the discussions mainly focused on the information needs of the other teams’ product managers. The operations team was convinced that understanding how their part of the platform or application is being used, can help product managers in taking a more targeted approach at assigning resources and making decisions on areas for im-

\(^1\)https://azure.microsoft.com/services/devops/
\(^2\)https://azure.microsoft.com/services/devops/repos/
\(^3\)https://azure.microsoft.com/services/devops/pipelines/
\(^4\)https://docs.microsoft.com/azure/azure-monitor/app/app-insights-overview
\(^5\)https://docs.microsoft.com/azure/azure-monitor/log-query/log-query-overview
\(^6\)https://docs.microsoft.com/azure/data-explorer/kusto/concepts/
\(^7\)https://azure.microsoft.com/services/developer-tools/power-bi/
provement and the evolution of their component. The available data comprised logins per user, uploads per user, page views by users on multiple levels of granularity (incl. timestamp and session id), service availability metrics, service performance metrics, user and customer data (e.g. registration dates), and sales data.

We started by getting an overview of the data, i.a. by looking at the fields, ranges and size of the data. In addition to that, we examined the distribution of metrics over time (see Figure 9-2a). Scripts for data cleaning and consistency checks were already setup by the operations engineer. Based on the discussions with the operations team, we came up with a couple of ideas for potential use cases.

**Use Case I: Navigation Path Patterns** As first use case, we investigated the navigation paths that users commonly use to navigate through the platform and its applications. In B2B contexts, it is important to differentiate between user-generated and customer-generated data. The usage data was only available on a user level since the customer is not the one using the product. Therefore, the navigation path analysis was only performed on a user level. More precisely, we selected the page views ordered by session id and timestamp to extract the users’ navigation flows. Figure 9-2b visualizes the navigation paths of one specific application as a directed graph. The nodes represent individual pages of the application while the width of the edges indicates the frequency of a chosen path. By looking at the graph visualization, stakeholders can determine between which pages users navigate to and from particularly frequently or only rarely.

In a next step, we presented the results to the operations team and asked them to provide feedback and additional ideas. While they liked the intuitive visualization of the navigation paths, they also noted that the session information gets lost. They would additionally like to know which pages are commonly called together within a session.

**Use Case II: Session Clustering** In order to address this, we applied a clustering technique to the same data set that clusters the individual sessions by the similarity of pages called within that session. The resulting clusters describe session types that consist of similar sets of page views. Figure 9-2c shows a subset of the results in
a sankey diagram. The left side represents the individual pages that can be called during a session. The right side presents the session types as clusters, while the connections between the two indicate which pages are often called together within the respective session types.

The results were again presented to the operations team in order to collect feedback. The discussions quickly evolved around the questions of how to use the results and what actions can be initiated based on it. According to the operations team, this will be the decisive factor in convincing other stakeholders as well. As a result, we conducted a joint brainstorming session in order to identify use cases in which the existing analyses can support stakeholders in taking informed decisions based on data. One idea brought up by the operations team was the combination of product usage and sales data to see if conclusions can be drawn from the usage data of a customer’s users to the purchasing behavior of that customer. Predicting customers that are potentially going to churn based on their users’ behavior could support the sales team to use their resources in a more targeted manner. By proactively contacting customers with a high risk of churn, the sales team can specifically take action to avert or minimize this risk.

Use Case III: Churn Prediction based on Usage Behavior  Therefore, we jointly defined the prediction of customer churn based on the product usage of a customer’s users as the next use case to further investigate. To achieve this, we first identified and extracted the use case specific data out of all available data sources. We selected a supervised machine learning technique to train a prediction model for our use case. While the model was trained on a customer level ($n = 2700$), we used both customer-generated and user-generated data to generate its input features. The customer-generated input features included the average uptime, downtime, and availability of a receiver installed at the customer sites, the number of affiliated users and number of days between registration date and license effective date. On the other side, the user-generated input features comprised the average time between registration and affiliation, average time since registration, average number of logins per day, sessions per day, sessions per type (derived from previous analysis), and uploads per day. The label (churn/non-churn) was extracted on a customer level based on the sales data. Since the operations team wanted to know more about the impact of behavior in different phases of a customer’s lifecycle, we divided the usage data into customer phases (e.g. “onboarding”, “usage”, and “pre-decision”) based on their timestamp. During preprocessing, we removed all rows with missing values and standardized the remaining input data. In a next step, we conducted a correlation analysis and applied a feature selection technique to the data set in order to identify and extract the most decisive input features which are used for training and testing the prediction model. To achieve this, we split the input data set into a training (80%) and a test set (20%). The training data was used as input for training the prediction model using a neural network. After a couple of iterations to optimize the hyperparameters and selected input features, the model achieved an accuracy of 88% on the unseen test set.

After presenting the results to the operations team, they provided positive feedback
and wanted to include the analysis in the dashboard which they were currently building for the sales team. In addition to that, the collection of additional data sources related to service reliability was discussed and implemented. We conducted joint sessions with the operations engineers and the operations manager to examine and discuss the newly collected data. Towards the end of the collaboration, the operations team considered data scientists as an integral part of their team as opposed to a detached entity running analyses for them.

**Collaboration Findings**

In summary, the stakeholders were already very open-minded towards adopting a DataOps mindset and data-driven ways of working in their team. They felt very comfortable in bringing in their own ideas, in giving feedback and in asking for additional analyses to implement. In this case, it worked really well to aim at making them feel involved as much as possible and at including them in any decisions on the use cases or analyses. Towards the end of the collaboration, the team had fully integrated data engineers and data scientists in their team. They were looking for a close collaboration in order to continuously improve or come up with new use cases. Table 9.2 summarizes the approach in Case A along two dimensions: activity categories (collaboration, data engineering, data analysis) and stages (preparation, execution, evaluation). Each activity can either be conducted by the data scientist (circle icon), by the product team’s stakeholders (triangle icon) or both (circle and triangle icon).

**9.4.2 Case B: Industrial Platform**

In Case B we collaborated with a platform provider operating in the industrial domain. Specifically, we cooperated with one product manager and one software architect who are responsible for tracking platform and app usage logs and making the data available to the individual product teams. In addition to that, we worked with two members of the platform’s sales team who are aiming at adopting a more data-driven way of working.

**Initial Situation**

Similar to the previous case, the platform of Case B is developed by distributed teams focusing on different components of the platform. In addition to the platform teams, there are company-internal as well as external partners who develop applications that are hosted on the platform. The platform teams were following DevOps principles for the development and operation of their platform. One essential part of DevOps is monitoring operational data in order to enable continuous feedback cycles [Erich et al., 2017] [Leite et al., 2019]. The platform provider setup a dedicated team for logging, processing and storing all requests users make to the platform and its applications. Their goal is to enable the other teams to extract continuous feedback based on the usage of their platform component or application. Even though they were still at the very beginning, the team has started to establish a DataOps mindset and is trying
<table>
<thead>
<tr>
<th>Preparation</th>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Regular sessions with stakeholders to assess and discuss information needs ○ △</td>
<td>• Multiple data sources available in common data storage (usage data, sales data, master data) △</td>
<td>• Method selection based on use case and data (supervised machine learning) ○</td>
</tr>
<tr>
<td></td>
<td>• In beginning: use cases [navigation paths] defined by data scientist based on stakeholders’ information need ○ → Later: joint definition of use cases (e.g. churn analysis) ○ △</td>
<td>• Scripts for data cleaning and consistency (already setup by operations engineer) △</td>
<td>• Method feasibility check taking B2B specific characteristics into consideration (e.g. analysis on customer level vs. user level) ○</td>
</tr>
<tr>
<td></td>
<td>• Use case specific data evaluation and identification of customer-generated and user-generated data related to the specified use case (e.g. labeling of customers based on sales data, examination of data related to quality of service and user behavior) ○</td>
<td>• Generic data evaluation (e.g. looking at fields, ranges, size, distribution etc.) ○, later on also in collaboration with stakeholders ○ △</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Extraction of use case specific data ○</td>
<td>• Implementation of navigation path analysis ○</td>
<td></td>
</tr>
<tr>
<td>Execution</td>
<td>• Intermediate discussions of preliminary results and work in progress during bi-weekly meetings ○ △</td>
<td>• Aggregation of raw data to input data sets (e.g. calculation of path frequency or average number of logins per day) ○</td>
<td>• Implementation of page view clustering ○</td>
</tr>
<tr>
<td></td>
<td>• Data preprocessing (e.g. removal of rows with missing values, standardization etc.) ○</td>
<td>• Data preprocessing (e.g. removal of rows with missing values, standardization etc.) ○</td>
<td>• Feature selection (e.g. for churn analysis) ○</td>
</tr>
<tr>
<td></td>
<td>• Implementation of churn analysis ○</td>
<td>• Iterative optimization of parameter settings and input data ○</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td>• Presentation and discussion of analysis results ○ △</td>
<td>• Collection of additional data sources related to service reliability △ → joint examination and discussion among data scientist and operations team ○ △</td>
<td>• Evaluation of analysis results (e.g. by calculating quality metrics) ○</td>
</tr>
<tr>
<td></td>
<td>• Stakeholders provide feedback and additional ideas (e.g. &quot;session information gets lost&quot; or &quot;need for more actionable insights&quot;) △</td>
<td>• Joint brainstorming and definition of use case adaptations and new use cases ○ △</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.2: Summary of *Case A* (○ = data scientist/data engineer activity, △ = stakeholder activity)
to get other teams on board as well. Since the platform is based on AWS\(^8\), the usage log team was using available services and storage solutions for processing and storing the incoming logs. Each request must pass through a gateway, generating a log entry each time. The log entry includes the timestamp, request url, user id, affiliated customer id, request size, response size, and a few other product-specific fields. As the amount of data being generated per day was quite large (100 GB), the usage log team stores historical data in an aggregated and compressed format while live data is continuously coming in.

**Implementation & Collaboration**

Two of our authors supported the usage log team with data engineering and data analysis tasks. For the duration of our ongoing collaboration, the platform provider gave us access to an anonymized copy of the usage logs as well as the sales data that tracks the base plans and add-on applications and services purchased by each customer. Since the usage log team wanted to separate any data analysis activities from the actual product, our first task was to set up a data analysis infrastructure for processing and analyzing data.

**Use Case I: Data Analysis Infrastructure** To achieve this, we set up a dedicated AWS account for this initiative that meets all security requirements. In order to copy the usage logs into that account, we set up two data pipelines that execute scripts for copying, processing, and storing the historical data and live data. The pipeline for the historical data was executed once to load all data from February 2019 to April 2020. Starting from that point, the second pipeline continuously collects the incoming live data over the time period from April 2020 to present. The usage logs are stored in a compressed format in AWS S3 buckets\(^9\) (26 GB per day). In order to have everything in the same place, we also moved the usage data and the sales data to the same AWS S3 storage (in separate buckets). Following this, we began to evaluate both data sources individually to get a better understanding of it. The sales data comprised fields for product information (e.g. product id, product name), customer information (e.g. customer name, sales id, reorder id), and date information (e.g. start date, expiration date, deletion date). In total, the sales data contained 11127 purchases of which 1307 were related to the actual platform accounts while the rest corresponds to upgrades, extensions and applications. As the product names for the same product were often inconsistent, we applied data normalization to the product name column to have a fixed set of available products. Moreover, the data contained duplicate rows which we removed during data cleaning. We also recognized missing values in the date column which we kept in mind in case we needed to filter these rows out for the upcoming analysis. In a next step, we started to explore the usage logs. The most relevant fields in the usage logs were the timestamp, request url, user id, affiliated customer id, request size, and response size. While querying the data in

---

\(^8\)https://aws.amazon.com

\(^9\)https://aws.amazon.com/s3
AWS Athena\textsuperscript{10}, we ran into a data type inconsistency. The data type of the numeric field \textit{request size} contains both integer and string values. Therefore, we setup a script that converts the data type of all string values to integer to enforce consistency.

**Use Case II: Churn Prediction based on Usage Behavior**  When the platform’s sales team became aware of this research initiative, they proactively approached us and expressed their interest in participating in this study. They were working together with the usage log team on building a dashboard that summarizes license information in combination with user activity in order to identify customers who are potentially not prolonging their license after expiration. However, the usage data was only selected from a gut feeling without evidence that it really relates to the purchasing behavior of customers. While the sales team wished for a more evidence-based indicator for this, there was a lot of uncertainty on their side about how to realize it. As a starting point of our collaboration, we setup a meeting with the sales team to get a better understanding of their situation. They were highly interested in shifting towards data-driven decision making but since they are not technologically trained, they struggled to know what is feasible from a data analysis perspective. In order to make this more tangible, we showed them the results of \textit{Case A} as a concrete example. The sales team saw value in the analysis and believed that a customer churn prediction model would help them in taking a more targeted approach at prioritizing the customers to be contacted. Therefore, we agreed to investigate the churn behavior of customers on a platform account level based on their users’ usage data.

To achieve this, we examined the data again in more detail with regard to the specific use case. First, we extracted the sales data for purchases related to the platform accounts \((n = 1307)\). Since we were interested in the customers’ decisions to prolong or cancel their subscriptions, we filtered out all accounts with expiration dates that occurred in the future as no decision has yet been made in these cases. As a result, we ended up with 518 sold platform accounts with expiration dates that lie in the past. Moreover, we grouped the additional purchase options (e.g. applications, extensions, upgrades) into four different categories and look at how many additional purchases were related to each of the platform accounts. During the sales data processing we regularly met with the sales team to discuss data inconsistencies and to clarify domain-specific questions on the data.

In a next step, we investigated the usage logs to identify metrics that can be used as input features for the prediction model later on. First, we aggregated the individual logs per user on a daily basis. As a result, the aggregated data set consisted of the daily active users, number of requests per day, number of requests per day and user, request size (minimum, maximum, mean, median, standard deviation), response size (minimum, maximum, mean, median, standard deviation), number of requests to the seven most common extensions and applications, and number of days to expiration date.

Analogous to \textit{Case A}, we selected a supervised machine learning technique to train the prediction model for our use case. While the model predicted the risk of churn on

\textsuperscript{10}\url{https://aws.amazon.com/athena}
a customer-level, we took both customer-generated and user-generated data into consideration. In order to build the input data set for the model, we used all expiration dates of each customer to label the 518 platform accounts that were extracted earlier (1 = churn, 0 = non-churn). Next, we added the number of additional purchases per category as input features. In addition to that, we added the aggregated usage behavior to each of the accounts. This step consisted of multiple iterations to find the most suitable representation. For one, we filtered out the last 90 days of usage before the expiration date because the sales team needs to know several weeks in advance which customers they should contact. Therefore, a prediction model that predicts customer churn only one day or one week before the customer's expiration date would not be of value to them. The remaining usage data was then further aggregated into customer phases of different lengths in multiple iterations, similar to Case A. In each iteration, we applied a feature selection technique and trained multiple prediction models with various classifiers. At the end of each iteration, we evaluated the models by looking the accuracy, precision and recall scores. Even the best performing model only achieved an accuracy of 72%. Current activities are ongoing in order to further improve the model.

Collaboration Findings

In summary, both, the technical and non-technical, stakeholders were interested in shifting towards a more data-driven way of working. However, they struggled to specify their information needs as they did not know what types of analyses could be applied to the data. Therefore, we aimed at supporting them in specifying their needs by explaining different types of analyses and the insights they can get out of it. Showing them results of previous use cases worked well in this case as it made the potential analysis more tangible for the stakeholders. They did not have to start from scratch but they could use that as basis to explain what they would be interested in and what should be different or the same in their case. Over the course of the collaboration, they became more confident in working with data and they started to bring in a couple of new ideas (e.g., additional data sources to look at). Table 9.3 gives an overview of the approach in Case B along the activity categories (x-axis) and different stages (y-axis). Each activity can either be conducted by the data scientist (circle icon), by the product team's stakeholders (triangle icon) or both (circle and triangle icon).

9.4.3 Case C: Visualization Application for Industrial Platform

In Case C we worked with an application provider whose application is hosted on the platform described in Case B. Specifically, we collaborated with the application's product manager.
### Preparation

<table>
<thead>
<tr>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Initial meeting to understand the stakeholders' problems and information needs ○ △</td>
<td>- Moving available data sources (usage data, sales data) to common AWS S3 bucket ○</td>
<td>- Method selection based on use case and data (supervised machine learning) ○</td>
</tr>
<tr>
<td>- Presentation of previous analysis → definition of churn analysis use case based on information needs ○</td>
<td>- Generic data evaluation (e.g. looking at fields, ranges, size, distribution etc.) ○</td>
<td>- Method feasibility check taking B2B specific characteristics into consideration [e.g. analysis on customer level vs. user level] ○</td>
</tr>
</tbody>
</table>

### Execution

<table>
<thead>
<tr>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Intermediate discussions to clarify domain-specific questions on data ○ △</td>
<td>- Aggregation of raw data to input data sets [e.g. number of requests per day, daily active users etc.] ○</td>
<td>- Feature selection ○</td>
</tr>
<tr>
<td>- Data preprocessing (e.g. removal of rows with missing values, standardization etc.) ○</td>
<td>- Implementation of churn analysis ○</td>
<td>- Iterative optimization of parameter settings and input data ○</td>
</tr>
</tbody>
</table>

### Evaluation

<table>
<thead>
<tr>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Presentation and explanation of analysis results ○ △</td>
<td>- Evaluating feedback and planning upcoming steps ○</td>
<td>- Evaluation of analysis result [e.g. by calculating quality metrics] ○</td>
</tr>
<tr>
<td>- Questions and feedback by stakeholders △</td>
<td>- Communicating next steps ○</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.3: Summary of Case B (○ = data scientist/data engineer activity, △ = stakeholder activity)
Initial Situation

The application is developed and released in an agile manner. The team has adopted DevOps principles but was not yet looking into data-driven feedback or data-driven decision making. Therefore, they currently saw no need for moving towards DataOps. However, since the application is hosted on the industrial platform of Case B, its usage data was already being collected by the platform’s usage log team. The usage log team would like to encourage individual teams, including the application’s development team, to make use of the data they collect for them. For this reason, we proactively reached out to the application’s product manager in order to introduce her to the idea and initiate a collaboration.

Implementation & Collaboration

During the initial conversation with the application’s product manager, she explained that she has been trying to push this topic a little bit but the priority was not high enough and there was no time and resources that can be allocated to the topic due to the pressure of further developing and improving the existing product. Even though she was skeptical that data-driven approaches were going to be adopted in her organization, she showed interest in the work we have done for the platform provider of Case A. Therefore, we presented her the results of the study and offered to run a similar analysis for her application. In order to better target her role as a product manager, we proposed to look into correlations and potential impact factors for customer churn that can support her in identifying areas for improvement within the application.

Use Case: Impact Factors for Customer Churn

The data sources for this use case were the same as for Case B as the application’s usage data is also tracked by the platform’s usage log team. We filtered both types of data by the application name. In the sales data we identified a total of 406 sold licenses for the application. 185 of these licenses were currently active and can, therefore, not be taken into consideration during our analysis. Another 76 rows needed to be filtered out due to variations of license types. Following this, the remaining licenses were labeled (1 = churn, 0 = non-churn) analogous to the labeling in Case B based on all expiration dates of the corresponding customer. In a next step, we extracted the following web usage metrics out of the application’s usage log data: daily active users, requests per day, requests per user and day, requests per type (based on request url), request ratio per type, requests per type and user, request size per type, response size per type, and minimum, maximum, mean, median, and standard deviation of request and response size.

We selected correlation analysis and supervised machine learning as method types for the analysis. Due to the data size limitations, we had to pay particular attention to the risk of overfitting and the significance of the results. In a first step, we aggregated the user-generated usage metrics on a customer level and added the label extracted from the sales data. Next, we standardized the input features and conducted a cor-
relation analysis on the data set in order to identify linear relations. The correlation coefficients indicated a low negative correlation between requests in specific areas of the application and customer churn. Following this, feature selection was applied to the data set to reduce the dimensionality and discard irrelevant features for training a prediction model. We used k-fold cross-validation to train prediction models with various classifiers (support vector machine, logistic regression, decision trees, and neural network). K-fold cross validation is often applied to small-sized data problems because instead of setting aside a validation set, the data set is split into $K$ parts. The model is then fitted to $K-1$ parts before calculating the prediction error for the $k$th part [Hastie et al., 2009]. This is repeated $K$ times until each part served as a test set in exactly one iteration. We applied k-fold cross validation to different classifiers and different parameter settings in order to identify the best performing classifier. After several iterations, the highest mean accuracy (78%) was achieved using logistic regression.

Before looking into the impact of input features on the prediction output, we presented the preliminary results to the product manager in order to keep her in the loop. More precisely, we showed her the results of the correlation analysis as well as the prediction model. The product manager was especially interested in the correlation analysis. While correlation does not equal causality, the results can be a first indicator of which areas within the application might be important to its users. As a product manager, she could not directly make use of the prediction model. However, it further sparked her interest in the potential impact factors for customer churn since it might enable her to detect areas for improvement within the application. As a result to her feedback, we proceeded with extending the analysis by additionally looking into the model’s feature weights and impact factors.

To achieve this, we used built-in attributes of the python packages used for training the models and additional python packages targeted to explain predictions. Scikit learn models have the `coef_` attribute which describes the “coefficient of the features in the decision function” \(^{11}\). In addition to that, we used the `eli5` package\(^ {12}\) to get a weighting of the input features ranked by their impact on the decision output. The results indicated that one specific area of the application was especially decisive. Both the requests to and response size of that area had a high impact on the prediction model’s outcome. We were not able to get the product manager’s feedback on this extension of the analysis since she left the company during the implementation phase.

Collaboration Findings

In this case, the stakeholder was rather new to the idea of adopting DataOps practices. While she was open-minded towards the collaboration, she stressed that her colleagues are very skeptical and cannot see the value in data-driven ways of working. As a result, it is difficult to allocate resources to the topic as it has no priority in the organization. In order to react to this skepticism, we had to be very proactive in reaching out to the stakeholder and in coming up with ideas. After the preliminary result presentation,

\(^{12}\)https://eli5.readthedocs.io/en/latest/overview.html
the product manager’s interest had increased and she asked us to proceed with the impact factor analysis. Table 9.4 summarizes the approach in Case C by activity category (x-axis) and stage (y-axis). Each activity can either be conducted by the data scientist (circle icon), by the product team’s stakeholders (triangle icon) or both (circle and triangle icon).

9.4.4 Case D: Advertising Analytics Platform

In Case D, we worked with an advertising analytics platform, specifically with the platform’s chief technology officer (CTO) and the vice president (VP) of product.

Initial Situation

The company has a background in building web site measurement tools within the customer data platform category so when building the new user interface a few years ago, the developers added very granular event tracking of all users using Segment.io13 to collect usage data. This usage tracking within the application included traditional metrics such as query response times and error rates together with feature-specific usage events such as "view report" and "download report". However, the dataset with millions of usage events collected was historically only used to calculate top-level metrics such as daily active users (DAU) or monthly active users (MAU) and aggregated manually on a monthly basis to review customer usage trends.

As part of an initiative to drive the adoption of DataOps across the organization and improve capabilities to proactively detect if customers were at risk of not renewing their annual contracts, the company launched an initiative, the Customer Scorecard. The goal of the Customer Scorecard initiative is to define a model which looks at all available metrics from the usage dataset, but also includes interaction metrics (customer emails, meetings, slack messages), support ticket metrics, engagement metrics (open rates of emails) and other available data sources that could contain relevant information about the perceived value of the product. The expectation from management was to get to a cross-departmental (sales, product and customer success) monthly review of each customer account to determine if any particular actions need to be taken.

Implementation & Collaboration

During the initial conversation, the platform’s CTO stressed his interest in shifting towards data-driven decision making. However, as they only had 19 active customers at that point in time, they were facing typical B2B-related data size limitations. Therefore, the platform provider was looking for a way to overcome this data size problem in order to gain more insights on their customers’ product usage.

Despite the limited number of customers, the platform provider had started to collect data from various sources which they made available to us. For one, they had implemented the event tracking functionality that logs page calls and events of the

13https://segment.com/
<table>
<thead>
<tr>
<th>Preparation</th>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Initial meeting with product manager to propose a collaboration and understand the stakeholders’ problems and information needs ○ △ (skepticism within organization towards adopting data-driven ways of working → no priority, time, and resources)</td>
<td>• Utilizing collected data of Case B stored in common AWS S3 bucket &amp; applying application-specific filters ○</td>
<td>• Method selection based on use case and data (correlation analysis and supervised machine learning) ○</td>
<td></td>
</tr>
<tr>
<td>• Presentation of previous analysis → use case definition based on information needs: identification of impact factors for customer churn ○</td>
<td>• Generic data evaluation (e.g., looking at fields, ranges, size, distribution etc.) ○</td>
<td>• Method feasibility check taking B2B specific characteristics into consideration (e.g., analysis on customer level vs. user level, apply k-fold cross validation) ○</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Execution</th>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Intermediate discussions to clarify domain-specific questions on application’s subscription types ○ △</td>
<td>• Aggregation of raw data to input data sets (e.g., number of requests per day, daily active users, request size per type etc.) ○</td>
<td>• Implementation of correlation analysis ○</td>
<td></td>
</tr>
<tr>
<td>• Data preprocessing (e.g., removal of rows with missing values, standardization etc.) ○</td>
<td>• Feature selection ○</td>
<td>• Implementation of churn analysis ○</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Presentation and explanation of analysis results ○ △</td>
<td>• Evaluation of analysis result (e.g., by calculating quality metrics) ○</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Questions and feedback by stakeholders △</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Evaluating feedback and planning upcoming steps ○</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Communicating next steps ○</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9.4: Summary of Case C (○ = data scientist/data engineer activity, △ = stakeholder activity)
customers’ users. For another, they use various tools in order to store and retrieve additional information. For example, they use Salesforce\textsuperscript{14} for monitoring their sales to customers. This tool offers the functionality to export sales data to csv files. Moreover, they use Zendesk\textsuperscript{15} as a ticket management tool which also offers an API to collect ticket data from code. For the email communication with customers (e.g. sending newsletters), the platform provider uses Mailchimp\textsuperscript{16} and Mandrill\textsuperscript{17}. These tools collect information on how actively the receiver of an email engaged with its content (e.g. opened email, clicked on link etc.). Both tools offer export functionalities to csv files. The platform provider gave us access to the usage data and the four tools listed above. In order to make use of the data, we initially had to come up with a concept for data extraction.

**Use Case I: Data Extraction** In a first step, we closely investigated the usage data in order to identify important events to focus on. The usage data was provided in a json file format. However, there was no predefined schema for the developers who are logging the event information. This resulted in a massive amount of key names as each developer came up with its own naming convention. One of our authors met with the platform provider’s head of product in order to identify important events to look for in the data. Following this, we spent a lot of time manually going through the data and mapping it to certain events. Once this mapping was complete, we setup a script that aggregates the usage data on a daily and a monthly basis per user. The daily data per user comprises a timestamp, user email, error ratio of events, and more than 60 product-specific events (e.g. when a user created, viewed, saved or exported something). The monthly data set is an aggregation of the daily data and additionally contains the fields user role (user/admin), number of active days, visit frequency, minimum recency, maximum recency, average recency, and error of days on which an error occurred.

The exported data from Mailchimp contained generic user-related fields such as email address, account, domain, and also member rating. The latter is a rating from one to five generated by Mailchimp on how actively the user engages with the content sent by the platform provider. The data exported from Mandrill provided some more detailed information on the email communication with users and customers. It contains the fields date, email address, sender, subject, opens, and clicks. The sales data was extracted from Salesforce which offers a broad functionality for generating custom reports. The sales data set we extracted contained the fields account name, region, effective date, live date, cancellation date, and days until live. Lastly, we extracted the ticket data per user by querying the Zendesk API from which we are able to get the number of opened tickets, closed tickets, ticket description, ticket priority, and ticket closing time. Due to the limited number of customers, we decided to first examine the data on a user level ($n = \text{814}$).

\textsuperscript{14}https://www.salesforce.com/
\textsuperscript{15}https://www.zendesk.com/
\textsuperscript{16}https://mailchimp.com/de/
\textsuperscript{17}https://mandrillapp.com/
Use Case II: User Classification

Initially we had planned to merge all available data sources in order to consider a wide variety of metrics. However, the overlap of timeframes of the individual data sources was too small in the beginning since the email data was only stored for the last three months. Therefore, we decided to proceed with the usage data and ticket only for now and to include the other data sources at a later point in time once there is more historical data available.

After merging the usage data and the ticket data, we applied a correlation analysis to the data set in order to identify linear correlations. The results indicated a high positive correlation between the number of tickets with priority normal and the ratio of error events and events that have taken longer than expected. In a next step, we applied k-means clustering to the data set in order to classify the users into different groups. The ideal number of clusters was determined by calculating the silhouette coefficient for different numbers of clusters ($k = \{2:10\}$). The lowest silhouette coefficient which resembles the best separation of clusters was achieved with $k = 5$ (see Figure 9-3a). As a result, the cluster analysis returned five clusters of usage behavior. Cluster 1 can be characterized by users with a low average recency, few events that have taken too long (threshold defined by stakeholders), and a low average time taken in general. Cluster 2 also has a low average recency, but high average time taken and many events that have taken too long. Cluster 3 describes users with a medium average recency, medium time taken, but many events of sharing artifacts with other users. In Cluster 4, the users have a medium recency and medium time taken but the error ratio is higher as compared to the other clusters. Lastly, Cluster 5 describes a high average recency, many events that have taken too long, and a high error ratio of events. In total, the clustering is based on 22 different dimension.

As an example, Figure 9-3b presents the clustering along one dimension: average time taken.

The preliminary results were presented to the CTO and the VP of product in order to collect their feedback and clarify some open questions on the data. The stakeholders liked the idea of describing each customer with a distribution of their users’ affiliated clusters. As an extension, they were interested in the changes per customer over time. Specifically, they proposed to run the analysis on a monthly basis and compare the results in order to identify a change in behavior.

Based on the stakeholders’ feedback, we extended the existing analysis and clustered the users per customer on a monthly basis. To visualize the progression over time, we created line plots for each customer representing the number of users per cluster and month. Figure 9-3c presents the line plot for one of the platform’s customers. Following this, the final results were presented to the CTO, VP of product and a newly hired data insights engineer. The stakeholders provided positive feedback and the data insights engineer is going to take over the analyses in order to integrate them into the Customer Scorecard.

Collaboration Findings

The stakeholders of Case D were already highly interested in driving the adoption of DataOps within their company. However, they were struggling to overcome B2B
related data size limitations and were uncertain how their data could be analyzed. In this case, it worked well for us to come up with and present a few ideas of how to process and analyze their data in order to react to their uncertainty. After seeing the first preliminary results, they stakeholders felt comfortable in giving feedback and bringing in their own ideas. Towards the end of the collaboration, they even hired a data insights engineer as part of the development team to specifically pick up this topic and integrate the results in their Customer Scorecard. Table 9.5 summarizes the approach in Case D along two dimensions: activity categories (collaboration, data engineering, data analysis) and stages (preparation, execution, evaluation). Each activity can either be conducted by the data scientist (circle icon), by the product team’s stakeholders (triangle icon) or both (circle and triangle icon).

9.4.5 Summary

In order to summarize the four cases presented in this paper, Table 9.6 gives an overview of common activities we conducted across all cases. Each activity was either conducted by the data scientist (circle icon), by the product team’s stakeholders (triangle icon) or both (circle and triangle icon). If an icon is placed in brackets, the respective role was not involved in the activity from the beginning but towards the end of the collaboration.
<table>
<thead>
<tr>
<th>Preparation</th>
<th>Collaboration</th>
<th>Data engineering</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Initial meeting to understand the stakeholders’ problems and information needs ○ △</td>
<td>• Multiple, distributed data sources available (usage data, sales data, ticket data, email data) △</td>
<td>• Method selection based on use case and data (k-means clustering) ○</td>
<td>• Method feasibility check taking B2B specific characteristics into consideration → analysis can only be run on user level, but results can be used to describe customers ○</td>
</tr>
<tr>
<td></td>
<td>• Data extraction from different tools (manually &amp; scripted) ○</td>
<td>• Evaluation of and concept for data extraction from tools ○</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Evaluation of and concept for data extraction from tools ○</td>
<td>• Generic data evaluation (e.g. looking at fields, ranges, size, distribution etc.) ○, later on also in collaboration with stakeholders ○ △</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Scripts for data cleaning and consistency ○</td>
<td>• Use case specific data evaluation and identification of customer-generated and user-generated data related to the specified use case (e.g. identification of important events) ○</td>
<td></td>
</tr>
<tr>
<td>Execution</td>
<td>• Intermediate presentation of preliminary results and clarification of domain-specific questions on data ○ △</td>
<td>• Aggregation of raw data to aggregates data sets (daily and monthly) ○</td>
<td>• Implementation of correlation analysis ○</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Extraction of use case specific data ○</td>
<td>• Implementation of usage data clustering ○</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Data preprocessing (e.g. removal of rows with missing values, standardization etc.) ○</td>
<td>• Iterative optimization of input data ○</td>
</tr>
<tr>
<td>Evaluation</td>
<td>• Presentation and discussion of analysis results ○ △</td>
<td>• Focus on usage and ticket data due to lack of overlapping timeframes ○</td>
<td>• Evaluation of analysis result (e.g. by calculating silhouette coefficient) ○</td>
</tr>
<tr>
<td></td>
<td>• Stakeholders provide feedback and additional ideas (e.g. “run analysis on monthly basis”) △</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9.5: Summary of Case D (○ = data scientist/data engineer activity, △ = stakeholder activity)
### Table 9.6: Summary of activities across all cases ($○$ = data scientist/data engineer activity, $△$ = stakeholder activity)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Activity</th>
<th>Case A</th>
<th>Case B</th>
<th>Case C</th>
<th>Case D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prep. / Coll.</td>
<td>Initial meeting to understand problems and information needs</td>
<td>$○△$</td>
<td>$○△$</td>
<td>$○△$</td>
<td>$○△$</td>
</tr>
<tr>
<td>Prep. / Coll.</td>
<td>Regular sessions with stakeholders to discuss information needs</td>
<td>$○△$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep. / Data</td>
<td>Merging data [e.g. in common data storage]</td>
<td>$△$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Prep. / Data</td>
<td>Generic data evaluation</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Prep. / Data</td>
<td>Use case specific data evaluation</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Prep. / Coll.</td>
<td>Use case definition</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Exec. / Coll.</td>
<td>Intermediate discussions or preliminary results and work in progress</td>
<td>$○△$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exec. / Coll.</td>
<td>Intermediate discussions to clarify domain-specific questions</td>
<td>$○△$</td>
<td>$○△$</td>
<td>$○△$</td>
<td>$○△$</td>
</tr>
<tr>
<td>Exec. / Data</td>
<td>Scripts for data cleaning and data extraction</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Exec. / Data</td>
<td>Data aggregation and data preprocessing</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Prep. / Analysis</td>
<td>Method selection based on use case and available data</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Prep. / Analysis</td>
<td>Method feasibility check [B2B specific data characteristics]</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Exec. / Analysis</td>
<td>Implementation of analysis</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Eval. / Analysis</td>
<td>Evaluation of analysis results</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Eval. / Coll.</td>
<td>Presentation, explanation and discussion of analysis results, incl. stakeholder feedback and questions</td>
<td>$○△$</td>
<td>$○△$</td>
<td>$○△$</td>
<td>$○△$</td>
</tr>
<tr>
<td>Eval. / Coll.</td>
<td>Planning of next steps based on feedback</td>
<td>$○△$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td>Eval. / Coll.</td>
<td>Definition or refinement of new/existing use cases</td>
<td>$○△$</td>
<td>$○$</td>
<td>$○$</td>
<td>$○$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$△$</td>
<td>$△$</td>
<td>$△$</td>
</tr>
</tbody>
</table>

9.5 Findings

After working with eleven stakeholders of four different product teams, we analyze and compare the cases of our study in order to gain insights on the incremental adoption of a DataOps mindset in B2B software-intensive companies. Our findings are structured into two categories: For one, we identify four stakeholder phases that the participants of our study went through during our collaboration. For another, we derive a generic model for driving the adoption of a DataOps mindset by collaborating with stakeholders, identifying use cases, and implementing analyses.

9.5.1 Stakeholder Phases

The mindset towards DataOps of the stakeholders in our study varies greatly. While some have not yet started thinking about DataOps at all, others are already in the middle of an organizational transition. Depending on how mature stakeholders and their organization are in that regard, the interaction with them is very different. We observe that the balance of initiative changes with the progress of adopting a DataOps mindset.

In the beginning of Case C, for instance, the application’s product team was not yet seeing the need for an organizational transition towards DataOps. Some of the product manager’s colleagues and managers could not see the value in data-driven ways of working which led to a lack of priority, time and resources assigned to the topic. As a result, we proactively reached out to them, came up with ideas, and implemented
a first proof of concept for them in order to spark their interest in the topic. The stakeholders in the early phases of Case B and Case D and the later phase of Case C, for example, were interested in data-driven decision making but were uncertain about what is possible from an analysis perspective. Due to this uncertainty, stakeholders often did not know what to ask of data scientists and data engineers. Therefore, our focus was on supporting and encouraging the stakeholders in specifying their information needs as well as explaining what types of analyses and insights are feasible based on the information needs and available data sources. In the later phases of Case B and Case D as well as in the early stages of Case A, the stakeholders started to bring in own ideas, give feedback on existing analyses, and ask for additional outputs and results. At this point, we aimed at making stakeholders feel involved by actively including them in decisions (e.g. how to extend analyses, which additional data sources to collect etc.). Moreover, we further encouraged stakeholders to give feedback and to express their additional information needs.

In the later phase of Case A and Case D, the stakeholders proactively brought up their own ideas and use cases. They also engage in and commit to a solid and permanent collaboration by fully including data scientists and data engineers in their team. On the other side, by being part of the stakeholders’ team, data scientists and data engineers participated in regular and continuous feedback sessions and continuously defined and discussed use cases in collaboration with the stakeholders. Based on these observations, we identify four stakeholder phases which are visualized in Figure 9-4.

Figure 9-4: Stakeholder phases during the process of adopting a DataOps mindset

Each column in the figure represents one phase and is divided into stakeholder phases
The first column describes the *skepticism* phase of stakeholders as observed in the early phase of *Case C* in our study. In this phase, stakeholders or their team members cannot see the value in data-driven ways of working and are, therefore, disinclined to assign priority, time, and resources to the topic. As a result, the lower part of the column describes *proactive engagement* as a way to react to the stakeholders’ skepticism. This includes proactively reaching out to the stakeholders, proactively coming up with ideas, and implementing a first proof of concept for them. The balance of initiative is uneven and considerably on the data scientists’ and data engineers’ side.

Once an initial interest is triggered, stakeholders move on to the second phase. This phase describes the later stage of *Case C* and the early stages of *Case B* and *Case D*. The stakeholders already show interest in the topic but do not yet have a feeling for what is possible from a data analysis perspective. The most common question stakeholders in this phase ask is usually “can you tell me how my product is being used?". From a data scientist and data engineer angle, it is therefore important to provide *explanations and inspiration* during interaction with stakeholders. This includes supporting and encouraging the stakeholders in specifying their information needs as well as explaining what types of analyses and insights are feasible based on the information needs and available data sources. The balance of initiative is slightly beginning to shift, but is still stronger on the data scientists’ and data engineers’ side.

Over time, stakeholders begin to get a feeling for specifying use cases and working with the analyses. As a result, they enter the *exploration and experimentation* phase during which stakeholders start to bring in their own ideas, proactively provide feedback, and ask for additional outputs and results. This was, for instance, perceived in the later stages of *Case B* and *Case D* and in the early stage of *Case A*. At this point, data scientists and data engineers can begin to strengthen the *inclusion* stakeholders. They should encourage stakeholders to give feedback and to express their additional information needs. Moreover, data scientists and data engineers can make stakeholders feel involved by actively including them in decisions. By further including the stakeholders in their work, the balance of initiative will be nearly even.

Once both stakeholders and data scientists and data engineers feel comfortable working together, they enter the last phase *collaboration*. They start to act as a team instead of two separate entities as observed towards the end of our collaboration in *Case A* and *Case D*. The stakeholders in this phase proactively bring up ideas and concrete use cases and commit to a solid and permanent collaboration with data scientists and data engineers as part of their team. On the other side, data scientists and data engineers continuously define and discuss use cases in collaboration with the stakeholders and participate in regular and continuous feedback sessions. After reaching the *collaboration* phase, the balance of initiative is completely even.

In general, stakeholders do not necessarily go through all of the four phases. In our study, only the stakeholder of *Case C* started out in the *skepticism* phase and moved towards the *interest* phase during the course of our study. The stakeholders of *Case C*...
9.5.2 Model Derivation

In a next step, we compare the entire process of the four individual cases from data preparation to use case specification, analysis, and stakeholder collaboration in order to derive a generic model for driving the adoption of a DataOps mindset. Since the interaction and collaboration with the stakeholders is different for each of the identified phases, we derive a base model which is then adapted to each of the stakeholder phases.

The starting point of the base model is a set of different data sources that are being collected for a specific product. The type of collected data sources is independent of the model and typically depends on the availability of data. The subset of data sources that were collected by all four cases in our study are usage data (e.g. in the form of log files) and sales data. In some cases additional data sources like master data records (Case A and Case D), or ticket and email data (Case D) are being collected. These data sources are merged into a common data storage, either by the authors supporting the individual teams (Case B, Case C, and Case D) or by the product teams themselves (Case A). In order to get a better understanding of the data, we conduct a generic data evaluation in each of the cases. This includes a close investigation of the available fields, ranges and distributions, the size of the data as well as the relations within and across data sources. Based on the evaluation, we proceed with the data preparation which, for instance, includes consistency checks, cleaning (e.g. removal of duplicate rows in Case B and Case C) and filtering (e.g. extraction of application-specific data in Case C). These two steps are usually part of an iterative process and it can take multiple iterations to reach a satisfactory level of data quality.

Either in parallel to or after getting this first overview of the data, we define a use case of the studied cases. The definition process highly depends on the stakeholder phase of the respective stakeholder. This is explained in more detail in the upcoming subsections presenting the adapted model in each of the phases. Once the use case is defined, we proceed with a use case specific data evaluation. As a common use case across the four cases we investigate customer churn based on product usage behavior. Therefore, we examine the sales data in greater detail. In particular, we consider the number of sales, apply labels (churn / non-churn) to historical sales data (with churn decision in the past), and look into the distribution of labels to detect imbalanced data. In Case A and Case D, for instance, the number of churning customers is either significantly lower or higher as the number of non-churning customers. This needs to be taken into consideration during the upcoming analysis. In Case B and Case C the number of churning and non-churning customers is almost balanced. Since we aim at investigating customer churn based on the products’ usage data, we additionally
need to identify and extract metrics out of the raw data that describe a customer’s or user’s interaction with the product (e.g. number of daily active users, average session duration etc.).

Based on the use case definition and the use case specific data, a suitable method type needs to be selected. In this process, B2B specific characteristics have a major influence on the selected method. For one, the scarce amount of data in B2B contexts can limit the type of applicable methods. For another, the complexity of having both customer-generated and user-generated data allows for different levels of analysis (customer level and user level). Moreover, a method’s applicability also depends on the type of input data (e.g. numerical, categorical, or text). During the course of this study, we typically aim at answering the following three questions when assessing the applicability of a method: 1) is the method feasible on a customer level or user level (depending on purpose or target of use case)?; 2) is the method feasible from a data size perspective (considering user and customer level)?; and 3) is the method feasible given the type of input data? If the answer to either of these questions is no, an alternative method needs to be selected following the same process. Taking the churn prediction use case as an example, this approach is very straightforward in Case A and Case B. The preferred method type for predicting customer churn based on product usage is a supervised machine learning technique. Since the customer is making the purchasing decision, the method is applied on a customer level. In these two cases, the amount of data does not constitute a problem for the selected method type and after extracting the identified metrics out of the raw data, they can be used as input data for supervised machine learning. In Case C, this process is more complex due to the limited number of customers. We still run the analysis on a customer level but we must pay particular attention to the risk of overfitting and the significance of the results. Therefore, we decide to apply supervised machine learning techniques using k-fold cross validation. In Case D the amount of data on a customer level is extremely low. Consequently, we cannot apply a supervised machine learning technique to the limited amount of data. With 19 customers it is not possible to draw a statistically significant conclusion. As an alternative we decide to run the analysis on an user level and use the results to describe each customer individually. More precisely, we select a clustering technique which is applied on a user level and is feasible from a data size and input data type perspective.

Before any analysis in each of the cases is implemented, a method specific data preparation is required. For example, in Case A, Case B, and Case C the usage data is available on user level but the analysis will be implemented on a customer level. Therefore, the users’ usage data needs to be aggregated on a customer level. Moreover, the input data is being standardized in all four cases. After the method specific data preparation, the analysis is being implemented. It may take several iterations to optimize parameter settings and selected input features.

After running the analysis, the quality and usefulness of results needs to be evaluated. If the quality of results is insufficient or if the results cannot be used to derive insights, the existing analysis is either being improved or an alternative method type needs to be selected. Otherwise, the results are prepared and presented to the stakeholders. Depending on the stakeholder phase, the way results are prepared, presented, and dis-
Figure 9-5: Base model for the adoption of DataOps practices in B2B software-intensive companies

cussed differs greatly which is explained in more detail in the following phase-specific subsections. The same also applies to the feedback evaluation. Based on the stakeholders’ feedback, the following questions are typically investigated, either by data scientists and data engineers or in collaboration with the stakeholders: 1) Which changes or improvements need to be made to the existing analysis?; 2) Which additional data input is required?; 3) Should the use case definition be revised or refined?; and 4) Are there any additional use cases? Based on the outcome of the evaluation, there are three ways to proceed. Either the existing analysis is revised in order to make the required changes or improvements, or data collection is adapted to collect additional input data, or existing or new use cases are defined. The described process derived from the four cases in our study is presented in Figure 9-5.

While the processes in the area of data collection, data processing, method selection, analysis execution, and analysis evaluation remain the same across all stakeholder phases, the collaborative steps (use case definition and refinement, results presentation, and feedback evaluation) depend highly on the stakeholder phases reflecting the stakeholders’ mindsets and organizational maturity of transitioning towards DataOps. The following subsections outline the instantiation of the model in each of the four phases, especially highlighting how the collaboration affects the individual steps.

Model in Phase I

In the first phase, skepticism, there is typically a lack of priority, time, and resources assigned to adopt data-driven ways of working. As a result, there is an increased need for proactive engagement by data scientists and data engineers. This imbalance of
initiative is also reflected in the adapted Phase I model variant presented in Figure 9-6. The colored boxes in the figure highlight the steps that are phase specific and extended from the base model. The icons indicate whether the steps are performed by the stakeholders as future users (user activity) or the data scientists and data engineers (data analyst activity).

The generic data evaluation and data preparation are performed analogously to the base model. The use case definition, however, is mostly driven by the data analysts. In Case C, for example, the product manager is skeptical that data-driven approaches are going to be adopted in her organization. As a result, we propose to run the churn analysis including feature impacts for her specific product. In our experience, proactivity in this stakeholder phase reduces the likelihood of a negative response. Based on this, the steps from use case specific data evaluation to evaluation of the analysis results remain the same as in the base model. The preparation and presentation of results aim at raising the stakeholders’ interest and collecting their feedback. In the skepticism phase it is, therefore, especially important to build up trust and prepare the results in an easily understandable way. During the results presentation in Case C, for instance, we first visualize the distribution of data in various bar plots to give the product manager an understanding of what the data we work with looks like. In addition to showing the results, we consider it important to explain the method and what to expect from it (e.g. how to interpret correlation coefficients). It is also helpful to clearly explain the input and output of an analysis (e.g. what are the input features, how are labels generated, what is the output). The goal of these explanations is to spark the stakeholders’ interest and to make them feel more comfortable in asking questions, providing feedback, and expressing additional needs.
The feedback evaluation in this phase is mainly conducted by the data scientists and data engineers. Based on the stakeholders’ feedback, they need to decide how to proceed. In Case C we choose to extend the churn prediction model by a feature impact analysis. Other possibilities include the adaptation of data sources or the definition of additional use cases or refinement of existing use cases. The latter is also driven by data scientists and data engineers.

**Model in Phase II**

The second stakeholder phase describes the state in which stakeholders show an interest in transitioning towards data-driven ways of working but at the same time they experience a lot of uncertainty with respect to specifying their information needs and the possibilities of analyzing their product-related data in general. Figure 9-7 presents the model adapted to the second phase interest. The colored boxes and icons represent the phase-specific steps which are either driven by the stakeholders as future users (user activity) or the data scientists and data engineers (data analyst activity). After the generic data evaluation and data preparation and analogous to the model in Phase I, the first phase-specific step in the model is the use case definition. In Phase II the use case is still being defined by the data scientists and data engineers. As opposed to use case definition in Phase I, this is typically done based on first discussions they have with the stakeholders in order to understand their interests and information needs. During these discussions, stakeholders should be supported and encouraged to specify their needs. In order to give them a feeling for what is possible, data scientists and data engineers should explain what types of analyses and insights are feasible taking the available data sources into account. In Case B the sales team
expresses a clear interest in adopting a more evidence-based way of making decisions. At the same time it is not their area of expertise to know what is possible from a data analysis perspective. As a result, we show them the results of Case A and propose to run a similar analysis for them. Since they are very keen on it, we proceed with the use case definition while taking their interests and information needs into account. In Case D the process for defining the use case is very similar. In the first discussion, the CTO highlights the company’s interest in becoming more data-driven but considering the limited number of customers he also experiences a level of uncertainty regarding data analysis and especially the statistical significance of results. Therefore, we take both his information needs and concerns into account for defining the use case.

The following steps from use case specific data evaluation to evaluation of results are inherited from the base model. The preparation and presentation of results is again phase-specific. Similar to the results presentation in Phase I, detailed explanations of how the results are generated should be given to the stakeholders. As part of the results presentation, a discussion is typically initiated in order to collect the stakeholders’ feedback. Thereby, the stakeholders should be encouraged to share their thoughts, ask questions, and express additional needs. On the other side, data scientists and data engineers can clarify questions they have on the data which require domain knowledge.

The collected feedback is consolidated and discussed by the data scientists and data engineers in order to plan the upcoming steps. These may include the adaptation of data collection, changes or improvements to the existing analysis, or the adaptation and definition of use cases. As opposed to the previous phase, the definition and refinement of use cases is a joint activity by data scientists, data engineers, and the respective stakeholders. Collaborating on the definition of use cases is a first step towards working as a team instead of two separate entities.

Model in Phase III

In the third stakeholder phase, exploration and experimentation, stakeholders proactively bring in their own ideas, feedback on a continuous basis, and ask for additional results. The extended model for Phase III is presented in Figure 9-8. Analogous to the previous extensions, the colored boxes and icons represent the phase-specific steps that differ from the base model.

In the third phase, the initial steps from data sources to generic data evaluation and data preparation are again adopted from the base model. While the use case is still mainly defined by the data scientists and data engineers, the stakeholders provide detailed input and clearly specify their needs. In Case A, for instance, we conduct multiple sessions for jointly looking at the data and discussing the product managers’ information needs prior to defining the use case.

The steps for implementing the use case (from use case specific data evaluation to evaluation of results) are replicated from the base model. Following this, the preparation and presentation of results is very similar to the one in Phase II. However, the discussions are even more in-depth and include brainstorming activities. In order to further improve the collaboration, it is important to make stakeholders feel included.
and to involve them in the decisions how to proceed. In the early phases of Case A and the later phases of Case B and Case D the stakeholders feel comfortable in providing continuous feedback, sharing their thoughts and asking for additional results. The results presentation is less structured as a formal presentation and discussion meeting but rather as a recurring working meeting among team members. For this reason, the feedback evaluation is a joint activity that is sometimes even part of the results presentation. During the evaluation, data scientists, data engineers and the respective stakeholders discuss how to convert the feedback into actions. Similar to the previous phases, this typically leads to either an adaption of data sources or to an adaptation of use cases or to changes and improvements to the existing analysis. In Case A for instance, after providing the feedback that the current results do not lead to concrete actions, the next steps are discussed together with the product team resulting in the churn analysis use case definition. In general, the following use case definition and refinement in this phase is performed as a team in a collaborative manner.

**Model in Phase IV**

The last stakeholder phase describes a state of mutual driven collaboration in which data scientists and data engineers are considered as part of the product team. The model adapted to Phase IV is presented in Figure 9-9. The colored boxes and icons highlight the phase-specific steps.

Since the stakeholders in this phase have typically already adopted a data-driven mindset, they are now additionally involved in the generic data evaluation. In joint
sessions, data scientists, data engineers and respective stakeholders examine the data, discuss inconsistencies and agree on what needs to be filtered. Towards the end of the collaboration in Case A, for example, the product team starts to collect additional data sources for measuring service reliability. We conduct a joint session to look at the data and discuss upcoming questions. Based on this, a generic data preparation is performed. The use case is jointly defined by the stakeholders as well as the data scientists and data engineers. The analysis implementation from use case specific data evaluation to running the analysis are adopted from the base model. Instead of a separate evaluation of results by data scientists and data engineers and the results presentation to the stakeholders, all involved parties jointly evaluate and discuss the results as a team as observed in Case A and Case B. They assess and debate the quality of results and the meaningfulness of insights that can be derived. Following this, the next steps are jointly discussed by the stakeholders, data scientists, and data engineers as a team. Depending on the outcome, they may decide to collect additional data sources, make changes or improvements to the existing analysis, or refine an existing use case or define a new use case. The latter is again performed as a team activity.

9.6 Related Work

Existing literature in the field of DataOps and specifically its adoption is still rather scarce. Raj et al. conducted a case study at Ericsson to identify different stages practitioners go through when evolving “from ad-hoc data analysis to DataOps” [Munappy et al., 2020]. The identified stages are: 1) Ad-hoc data analysis (created on demand for specific purpose); 2) semi-automated data analysis (e.g. including data pipelines, data technologies, and data processes); 3) agile data science (development & deploy-
ment of code according to agile and DevOps principles); 4) Continuous testing and monitoring (e.g. of data pipelines); and 5) DataOps (DevOps for data analytics incl. CI/CD, collaboration in mixed teams) [Munappy et al., 2020]. While the authors touch upon the interaction between data teams and their customers, their main focus lies on the technical evolution. Our study differs from that, as it is mainly concerned with the interaction and collaboration with stakeholders from a data engineer and data science perspective.

In addition to related literature in the area of DataOps, we identified a few related studies in the area of organizational transformation, trust in data analytics and machine learning, and data scientists working in software development teams. Organizational transformations that affect the way teams are being used to work, is considered to be a challenging task [Walls, 2013]. It can be difficult to prove the necessity of change since the “costs must be outweighed by the benefits” [Walls, 2013]. Moreover, also the management has to be convinced in order to facilitate organizational changes. This requires open communication which ultimately leads to an improved collaboration [Walls, 2013].

People’s trust increases over time if the observed performance of a machine learning model meets or exceeds one’s expectations and generates convincing results in practice [Yin et al., 2019]. Moreover, building up trust in a new technology, such as machine learning, is “a dynamic process, involving movement from initial trust to continuous trust development” [Siau and Wang, 2018]. Thereby it is important to design an analysis as transparent and explainable as possible in order to increase its interpretability and, consequently, also the trust in the technology and its results [Siau and Wang, 2018]. The results of our case study are consistent with the literature in this regard. Building up trust by providing explanations and supporting stakeholders in understanding and interpreting the results played a very essential part during the collaboration with the stakeholders in our case study.

Kim et al. [Kim et al., 2016] interviewed 16 data scientists at Microsoft to get a better of their backgrounds, problems and activities as well as their style of working with or in software development teams. They identify five different data scientist working styles: 1) Insight providers (work with engineers to get required data, analyze product and customer data to support managers’ decision making, iteratively collaborate with managers to comprehend and refine goals, communicate results to team); 2) model specialist (experts working with insight providers on specific types of analyses); 3) polymaths (cover all tasks from business goal specification, data collection, running analyses, and communicating results); 4) platform builders (design and build reusable platforms, e.g. for collecting and storing data) 5) team leaders (lead team of data scientists, push “for the adoption of data-driven decision making" [Kim et al., 2016]). In a more recent study, Kim et al. [Kim et al., 2017] additionally conducted a large-scale survey among data scientists in order ot get a broader picture of how data scientists work in a software engineering context. Based on the data scientists’ tasks and activity, the authors cluster the respondents into nine data scientist clusters: Polymath, data evangelist, data preparer, data shaper, data analyzer, platform builder, fifty-percent moonlighter (spending only 50% of their time on data science tasks), twenty-percent moonlighter, and insights actors [Kim et al.,
2017]. The way we have worked with the product teams in our case study overlaps to some extent with the working styles identified by the authors, especially the insight provider. However, their work covers the ways of working after data scientists are already being part of software engineering teams. Our case study focuses on the transition from traditional software engineering teams to mixed teams (including data scientists or data engineers) as part of DataOps.

### 9.7 Threats to Validity

Two of the cases in this study originate from the same company (*Company A*) and a third case from a subsidiary company of *Company A*. This might constitute a threat to the external validity of our study. However, all the platform and application providers operate in very different domains and in order to further compensate this, a fourth case from a different company was included in the study.

One threat to the internal validity of our study is the difference in length of the collaboration periods across the four cases due to availability and prioritization on the topic. This results in a somewhat imbalanced amount of empirical data collected for each case. While our study covers a variety of different roles and domains, the majority of the stakeholders participated due to their interest in DataOps and data-driven ways of working which might bias the results to some extent. Nonetheless, we believe that the presented findings provide value to practitioners who are trying to drive the adoption of DataOps in their organization.

### 9.8 Conclusion

A transition towards DataOps can enable software-intensive companies to make more meaningful use of their data. It can be considered an enabler for continuous data analysis and machine learning applications. Data-driven insights can be generated to provide fast feedback for continuous improvement. However, while the benefits of adopting DataOps practices are manifold, the adoption process is often a difficult journey as it entails organizational changes (e.g. mixed teams of software developers and data engineers / data scientists), a shift of mindset (e.g. making decisions based on data instead of opinions / gut feelings), and a certain level of trust and confidence (e.g. believing in the value of data-driven ways of working).

In order to investigate this journey, we conducted an exploratory case study in four real-world product teams across two B2B software-intensive companies. Our findings indicate that there are different phases stakeholders go through during their journey of adopting DataOps practices. The stakeholder phases are called skepticism, interest, exploration/experimentation, and collaboration. Depending on the stakeholders’ phase, data engineers and data scientists need to react differently according to the stakeholders’ needs. As a result the balance of initiative is highly imbalanced towards data engineers and data scientists at the beginning but balances out later on.

Moreover, we derived a generic model that covers the process steps from all four cases.
across three stages: preparation, execution, and evaluation. In order to meet the individual stakeholder needs, the model is adaptable to each of the identified stakeholder phases.

The results of our case study indicate that it is crucial to adapt different steps in the adoption process to the stakeholders’ needs and mindsets. The stakeholder phases highlight how data engineers and data scientists can react to different types of stakeholders who require different types of interaction, communication and collaboration. Practitioners who are trying to push for the adoption of a DataOps mindset can use the presented stakeholder phases and corresponding model to tailor the adoption process to their team’s maturity with regard to DataOps and data-driven ways of working.

Currently our study is limited to a short-term view on the adoption journey with an average collaboration period of one year. In the future, we aim at extending this study by more long-term experiences and additional cases.
Chapter 10

Conclusion

This chapter concludes the thesis and is structured as follows. First, the overarching research questions defined in Section 1.2.2 are addressed and answered in Section 10.1, before the threats to validity are presented in Section 10.2. Following this, the generalization of this work is discussed in Section 10.3. Finally, the key contributions of this thesis are highlighted in Section 10.4 and an outlook of further research is given in Section 10.5.

10.1 Research Questions

The following subsections describe how the studies address the research questions presented in this thesis. In addition to that, Figure 10-1 is an adapted version of Figure 1-1 indicating how the research studies (boxes with rounded corners) build upon each other and highlighting the key results (boxes with sharp corners) of the research studies mapped to research questions. Some key results can be mapped to exactly one research study (boxes are overlapping), while others are derived from multiple research studies (box is placed on arrow connecting the respective studies).

RQ1: What are the key challenges that teams in B2B software-intensive companies face when coordinating their efforts, making decisions, and exchanging data and feedback?

This research question was addressed by Chapter 2 of this thesis.

Conflicting interests constitute a major reason for challenges. One key finding of this study is that conflicting interests within and across teams and organizations are a major reason of (perceived) challenges. For example, the platform providers experience difficulties in prioritizing feature requests from their partners (customers) as they receive a large number of poorly defined requests. The partners, on the other hand, complain about a lack of communication and inconsistent processes. As a result, this often leads to misunderstandings which causes displeasure on both sides.
Another example is that the platform provider and its partners would like to receive as much information on their customers as possible in order to make informed decisions on product evolution. In complex B2B setups, they often have to rely on others to share certain information or feedback as they do not have direct customer access. However, many teams are disinclined to share data, information, or feedback with others especially across teams or organizational borders, e.g. due to additional effort and no direct benefits for them. Additionally, there are no established processes or infrastructure in place to improve this.

**Key Challenges.** In total, nine challenges were identified during the course of the study. Three main key challenges – coordinating efforts, making decisions, and exchanging data and feedback are highlighted below:

- **Establish appropriate communication:** One of the key challenges in coordinating the teams’ efforts was to establish appropriate communication channels and processes on topics and deliverables. This was mostly caused by long communication paths, many different communication channels, and a lack of well-established processes.

- **Decision making:** Another challenge affecting decision-making processes is the difficulty to obtain a broad picture of all customers in order to plan and prioritize
upcoming development activities. This is often caused by the fact that it is difficult to get direct customer access in B2B contexts. Oftentimes, there are restrictions on who is allowed to talk to the customer.

• Exchange of data and feedback: A third key challenge is the lack of established processes to collect and exchange data and customer feedback. There exist no direct feedback channels to store and access feedback. Depending on who collects feedback from customers, information might get lost or altered and the feedback might be very one-sided. While it is beneficial for teams to receive data and feedback from others, there is often no direct advantage in sharing information with others. Therefore, teams are often not inclined to share data and feedback except when there is a direct benefit in it for them.

RQ2: How can operational data with B2B-specific data characteristics be leveraged to generate technical and business insights for different roles involved in the product lifecycle?

The research questions RQ2 was addressed by four research studies in this thesis: Chapter 3, Chapter 4, Chapter 5, and Chapter 6.

Incremental analysis approach for generating high-level insights from low-level operational data. In Chapter 3 and Chapter 5 of this thesis, operational product and customer data was analyzed to generate and measure insights for multiple different roles (product managers, software architects, operation engineers, and sales). Based on the implementations and the interaction with the stakeholders, a generic model was derived to incrementally process and analyze quantitative data to generate stakeholder-targeted insights on three different levels: operational, tactical, and strategic. These levels constitute one dimensions of the model and originated due to different types of questions that stakeholders had during the course of the studies:

1. Operational: aims at answering immediate questions ("what?") and providing basic insights by using solely low-level metrics to describe a current state or situation (e.g. comparison of current feature usage rates); The goal is to provide fast feedback of current behavior

2. Tactical: aims at interpreting data ("how?") and describing certain phenomena, behavioral patterns, or unexpected behavior by combining low-level metrics with the previously gained knowledge and applying advanced techniques (e.g. clustering of APIs that are often used in combination); The goal is to identify and continuously monitor the current business or customer value, to enable product managers to base their decisions on detected behavior, and to provide an understanding of how previous decisions influenced the current value

3. Strategic: aims at utilizing the data in order to provide further suggestions, explanations or interpretations that are beyond just a tactical representation of data ("how to...?"); the goal is to increase the business or customer value in the
future by providing continuous suggestions for optimizations or concrete action items to increase business or customer value (e.g. website structure optimization based on navigation flows)

The second dimension of the model describes the process of getting from input data to knowledge within each of the levels (input data → planning → execution & interpretation → knowledge). The key concept behind the model is that the individual levels build upon each other. More precisely, the knowledge gained in one level is used as additional input data in the subsequent levels. Thereby, the complexity of information that can be generated is incrementally increased, starting from operational data and ultimately leading to high-level insights.

**Primary data sources and reuse of analyses across stakeholders.** Chapter 5 of this thesis investigates how the approach of Chapter 3 can be efficiently implemented for different types of stakeholders. One observation that stuck out during the implementation was the relevance of different types of data sources to the individual stakeholders. As a result, a set of primary data sources could be identified for each stakeholder. These are not set in stone and can be adapted to the evolving needs of stakeholders. However, the purpose of assigning primary data sources to a stakeholder is to narrow down the problem space in order to generate results faster and more efficiently.

Another observation made during the implementation of the individual cases is the opportunity to reuse different components across stakeholders. Analyses that were originally made for a different stakeholder, can often be reused for other stakeholders as well (e.g. API usage clustering or usage path analysis), especially in the strategic level. This results in a data and knowledge flow across stakeholders increasing the efficiency of implementations. Ultimately, the model presented as a result in Chapter 5 presents an iterative process for implementing stakeholder-targeted monitoring and decision-making frameworks. The model fosters the sharing and reuse of knowledge and components, tools, and concepts to speed up the instantiation of the process. The process itself is based on a set of primary data sources and methods that were identified for each stakeholder during the analyses.

**Handling B2B-specific data characteristics.** Dealing with B2B-specific data characteristics was a major concern in all four of the chapters addressing RQ2. Especially Chapter 4 and Chapter 6 closely investigate how B2B-specific data characteristics can be taken into consideration when processing and analyzing customer and product data. In both research studies, the volume of data was quite low (approx. 1400 - 2000 customers). Moreover, both customer generated data (e.g. sales data) and end-user generated data (e.g. product usage data) had to be considered during the analysis. In order to run an analysis on customer level (e.g. churn prediction or customer satisfaction), the end-user generated data was aggregated and mapped on customer level. In Chapter 4 of this thesis, relevant customer phases were identified in order to create a time scale for aggregating customer and end-user data. Different types of metrics were considered during the analysis: static metrics (e.g. number of
days between registration date and license effective date) and dynamic metrics (e.g., average number of sessions per day). Moreover, each metric can either be a direct metric (e.g., license type) or a derived metric (e.g., average number of sessions per type). The mapping of metrics to the identified customer phases generates a data set for further analysis (e.g., customer churn prediction) while taking B2B-specific data characteristics into consideration.

B2B-specific characteristics do not only influence the type and amount of data but also the type of use cases to be investigated. For example, it is difficult to assess customer satisfaction in B2B contexts without having direct access to end-users. Therefore, there is a need for measuring customer satisfaction in a fully quantitative way without requiring active customer or end-user participation which was investigated in Chapter 6 of this thesis. Customer satisfaction was measured by using service quality and web usage metrics. The service quality metrics were only available on a customer but not on an end-user level. As a result, the customer satisfaction score formula only considers customer-generated data. The web usage metrics, on the other hand, were only available on an end-user level as the customer is not the one actually using the product. Therefore, a similar mapping as in Chapter 4 had to be applied to aggregate the web usage metrics on a customer level. One particularly notable outcome was that the results of the customer satisfaction score formula (only customer generated data) somewhat correspond with the results of the web usage metric classification (only end-user generated data).

RQ3: What are the challenges on a technical, organizational, and individual level in operationalizing analyses for actual use in practice and how can these challenges be addressed?

This research questions was addressed by Chapter 7, Chapter 8, and Chapter 9 of this thesis.

The vicious circle impeding the operationalization of analyses. Chapter 7 of this thesis investigates the challenges on a technical, organizational, and individual level in productizing analyses in the context of data-driven software engineering. Both a qualitative interview study and a quantitative survey was conducted to derive a list of challenges that practitioners face in practice. The results indicate that there are five interdependent key drivers forming a vicious circle that impedes the operationalization of analyses. The key drivers are: 1) Lack of priority, time, and resources; 2) low data quality; 3) inability to cross cultural gap; 4) ineffective prototypical analysis; and 5) inability to prove value. In summary, the low priority assigned to the topic results in two consequences: On the one side, an inadequate amount of time is spent on dealing with data quality. On the other side, the discussions between users and providers of AI-based analytics solutions in the context of software analytics and business intelligence are not sufficient to overcome communication problems, create a mutual understanding, and define relevant use cases. This cultural gap, along with poor quality of data, leads to prototypical implementations whose results cannot deliver the desired value. In turn, if the value is not clearly visible, it is challenging
to convince others of that value. So the vicious circle is complete as this again will result in not increasing the priority of the topic.

**Overcoming technical difficulties.** In Chapter 7 of this thesis, potential solutions to the technical challenges are derived from the interviews, survey results, and existing literature. Practitioners recommend to push for a common definition and a centralized storage of data to make its use more efficient. Moreover, a unified data model can help to make data sources easily extensible and connectable as they all share the same level of granularity.

In addition to that, Chapter 8 of this thesis examines the technical process of getting from a prototypical ML-based analysis to an automated and deployed analysis. Existing literature on data management and processing, model building, and model deployment was scanned to derive a framework that comprises all key activities from data collection to retraining of deployed models. In addition to that, the framework is structured in three iterative cycles: a prototyping cycle, deployment cycle, and an update cycle. These cycles resemble stages in the lifecycle of a ML model and by outlining the transitions between stages, the framework specifically guides the journey from a prototypical analysis to a productively running ML model.

**Driving a data-driven mindset in B2B software-intensive companies.** Some of the organizational and individuals-related challenges often originate from a lack of data-driven mindset established in individual teams and entire organizations. Chapter 7 presents potential solutions to the challenges based on the interview study, survey results, and existing literature. A close and continuous collaboration between data scientists and stakeholders is inevitable. Use cases need to be defined in an iterative manner with regular feedback sessions to support the interpretation of results and identify problems faced by the stakeholders. Explanations by data scientists to stakeholders should be kept simple and, if possible, even be translated into the stakeholder’s domain instead of using data science measures. Moreover, it is important to discuss, define, and explain what actions can be taken based on the results.

Chapter 9 of this thesis further investigates how data-driven mindsets are adopted in B2B software-intensive companies. The findings indicate that there are different phases stakeholders go through during their journey of adopting data-driven practices. The stakeholder phases range from skepticism over interest over exploration/experimentation to collaboration. Depending on the stakeholders’ phase, data engineers and data scientists need to react differently according to the stakeholders’ needs. As a result the balance of initiative is highly imbalanced towards data engineers and data scientists at the beginning but balances out later on.

**10.2 Threats to Validity**

This section discusses the threats to validity structured by the three research questions of this thesis. To address risks to *construct validity* in all studies of this thesis, all interview guides were designed with open questions. The case study descriptions
contain background information on both the case products and the interviewees to give as much context as possible. Observer triangulation was used to mitigate the observational bias. One to two researchers were involved in conducting the interviews, while four researchers from different organizations contributed to the research design and analysis of collected data.

10.2.1 Threats to Validity Relevant to RQ1

RQ1 What are the key challenges that teams in B2B software-intensive companies face when coordinating their efforts, making decisions, and exchanging data and feedback? is addressed by Chapter 2 of this thesis. The study is based on qualitative interview data.

A threat to the internal validity of the study is the number of studied cases. This is partially mitigated by the fact that the platforms’ development is very large-scale (up to 750 developers) and distributed across a multitude of teams. Each interviewee participating in the case study belonged to a different and individual product team. Moreover, the interviewees were not randomly sampled as they were selected by contact persons of the respective platforms.

A threat to the external validity of the study is that the studied cases both originate from one company and a subsidiary of this company. However, the case company is very large and has multiple subsidiary companies. Therefore, the case platforms operate in two very different domains (industrial domain and healthcare domain) and are completely independent of each other.

10.2.2 Threats to Validity Relevant to RQ2

RQ2 How can operational data with B2B-specific data characteristics be leveraged to generate technical and business insights for different roles involved in the product lifecycle? was addressed by Chapter 3, Chapter 4, Chapter 5, and Chapter 6 of this thesis. The studies are based on qualitative interview data and quantitative measurement data.

Threats to the internal validity relevant to this research question are the number of cases per study in Chapter 4 and Chapter 6 as well as the number of cases per stakeholder type in Chapter 5. In order to partly mitigate these threats, the B2B-specific characteristics of the case in Chapter 4 was compared to characteristics identified in existing literature. Chapter 6 of this study follows a deductive research approach. Therefore, the model was derived from theories extracted from published literature and the case was used for validating the model. While multiple stakeholders of the same type participated in the study in Chapter 5 (e.g., operation engineers), they sometimes belonged to the same product team and are, therefore, only considered as one case even though multiple participants contributed to the case.

Similarly to threats relevant to RQ1, all cases originate from the same case company and its subsidiary. As indicated before, the case company is very large and consists of many divisions and subsidiary companies. This threat is, therefore, partly mitigated by the fact that the participants in the research studies presented in Chapter 3 and
Chapter 5 are distributed across several independent product teams with their own processes and ways of working.

10.2.3 Threats to Validity Relevant to RQ3

RQ3 What are the challenges on a technical, organizational, and individual level in operationalizing analyses for actual use in practice and how can these challenges be addressed? was addressed by Chapter 7, Chapter 8, and Chapter 9 of this thesis. The studies are based on qualitative interview data and quantitative survey data.

One threat to the internal validity in Chapter 7 is the limited number of products and the unequal distribution of interviewees among product teams. This was partially compensated by including the consultants who both work for the underrepresented product but also for a number of other products. Moreover, it is likely that the respondents of the survey participated due to their interest in the topic, thereby potentially causing a positive bias in the results. Due to the limitations of the survey tool, dependencies between questions and factors could not be taken into consideration. Due to the lack of successfully deployed analytics solutions, the group of experts interviewed for validating the solutions consists mostly of technical experts (providers) as compared to non-technical stakeholders (users). Therefore, a bias in the results due to the over-represented providers cannot be ruled out.

One threat to the internal validity in Chapter 8 is the validation of the update cycle. A more long-term validation is required to fully validate whether a retraining of the model was successful in practice. However, the activities presented in the different cycles of the model were derived from existing literature while the case was used for validation which partially mitigates the threat.

The limited number of interviewees in Chapter 7 might have an impact on the generalizability of the results which constitutes an external threat to validity. This is partially mitigated by using a mixed-methods approach and additionally conducting a survey to answer the same research questions. However, the same applies to the expert interviews conducted to validate the potential solutions. Furthermore, the transferability of results to other contexts might be affected by a positive bias in the survey results.

While the overarching case studies in Chapter 7 and Chapter 9 were conducted in multiple unrelated and independent product teams, the majority of cases originate within a similar industrial context which might impact the external validity of the studies. However, with the many roles, organizational units and products represented in the study and by multi-source data collection as the basis for the results, there is a certain level of confidence that the presented findings provide value also outside the specific context of the case study company.

10.3 Generalization

The results presented in this thesis are based on both qualitative (e.g. interviews) and quantitative (e.g. survey results) data collected from one or multiple case compa-
nies. Since the limited number of case companies might raise concerns regarding the
generalizability of results, it is important to highlight the characteristics and diversity
of the case companies. The majority of cases are based on case companies Company
A and Company A1 (see Table 1.3). Company A is a large, industrial company with
around 293,000 employees and multiple subsidiary companies (e.g. Company A1).
The case company and its subsidiaries offer a diverse product portfolio in a variety of
different domains, such as industry, infrastructure, mobility, healthcare, and energy.
It is an international company with product teams that are spread all over the world
(mostly in Europe, USA, India, and China).
Even though several cases originated from the same company, the twelve individual
product teams participating in the studies are very diverse. They are all operating in
very different domains, following their own processes, and being located in different
parts of the world. Except for sharing the same IT infrastructure, the product teams
are very independent of each other and have established their own ways of working
which supports the generalizability of results presented in this thesis.
Furthermore, an additional case company (Company B) was included in Chapter 9
of this thesis. While the other product teams mostly cover diverse but industrial
contexts (e.g. industry, infrastructure, mobility, healthcare, and energy), Company
B operates in the media domain as an advertisement analytics platform. In general,
this thesis specifically targets B2B software-intensive companies but its results are not
limited to one industry segment but are generalizable to several industry segments in
the B2B domain.
In terms of reproducibility, each chapter of this thesis comprises a detailed descrip-
tion of the data collection (either qualitative or quantitative), data processing, and
data analysis applied in each study. In the interview-based studies, interview guides
were prepared before conducting the interviews. Each interview was transcribed and
summarized afterwards, before applying thematic coding to the transcript. This was
used as a basis for extracting and describing the findings presented in the studies. In
studies that present the investigation of quantitative (e.g. product usage) data, the
fields and size of the data are described in a transparent way. In addition to that, the
algorithms and techniques applied to the data are named and referenced in the text.

10.4 Key Contributions

This thesis covers several aspects of adopting data-driven practices in B2B software-
intensive companies. It starts with an investigation of challenges that teams in B2B
software-intensive companies face when coordinating their efforts, making decisions,
and exchanging data and feedback. Following this, several conceptual models are
introduced for generating technical and business insights for a variety of roles (e.g.
software architects, product managers, or sales) based on operational (product and
customer) data, while taking B2B-specific data characteristics into consideration. Fi-
nally, a set of technical, organizational, and people-related challenges in productizing
(automating and deploying) prototypical analyses is identified and a subset of these
challenges is further investigated in order to examine and develop approaches for ad-
dressing the challenges.
As a result, the key contributions of this thesis are an overview of challenges in coordinating efforts, making decisions, and exchanging data and feedback in agile teams in B2B software-intensive, four models for systematically generating high-level insights based on low-level operational data, a vicious circle that highlights interdependent key drivers impeding the productization of analyses, a framework for the productive use of machine learning in software analytics and business intelligence, and a model for the adoption of a data-driven mindset in B2B software-intensive companies.
The specific contribution of this thesis with regards to the challenges in coordinating efforts, making decisions, and exchanging data and feedback in agile teams in B2B software-intensive companies is:

- The majority of challenges can be traced back to conflicting interests between different actors within and across teams and organizations. Nine conflicting interests are identified and presented as the source of the challenges.

- A set of nine challenges is derived including a detailed evaluation of why these challenges occur. Three key challenges with regard to coordinating efforts, making decisions, and exchanging data and feedback are: 1) Establishing appropriate communication paths for topics and deliverables (mostly caused by long communication paths, many different communication channels and a lack of well-established processes); 2) Difficulty to obtain broad picture of all customers in order to plan and prioritize upcoming activities (difficult to get customer access); and 3) Lack of established processes to collect and exchange data and customer feedback (no direct feedback channels, people are disinclined to share information with others).

The specific contributions of this thesis with regards to systematically generating high-level insights based on low-level operational data are:

- Stakeholders have different types of questions on different levels of complexity (operational, tactical, and strategic). Knowledge gained in one level is used as additional input data in the subsequent levels. Thereby, the complexity of information that can be generated is incrementally increased, starting from low-level operational data and ultimately leading to high-level insights.

- Not all data sources are equally relevant for all types of stakeholders. Depending on the information need, it makes sense to focus on a subset of data sources which can be adapted along the way. This helps to narrow down the problem space and to generate results faster and more efficiently.

- For higher-level insights (e.g. in the tactical or strategic level) there is often the opportunity to reuse data across different types of stakeholders. This results in a data and knowledge flow across stakeholders, also increasing the efficiency of implementations.

- Operational (customer and product) data in B2B contexts can either be customer-generated or end-user-generated. Relevant customer phases can be used as a
time dimension to aggregate and map the individual data sources. The resulting
data set can be used for further analysis, such as the customer churn prediction
in Chapter 4.

- B2B-specific characteristics do not only influence the type and amount of data
but also the type of use cases (e.g. if direct access to customers or end users is
not feasible). Chapter 6 presents an approach for measuring customer satisfac-
tion in a fully quantitative way without requiring active customer or end-user
participation. The analysis was implemented from two different angles: 1) using
customer generated data (customer satisfaction score formula based on service
quality metrics); and 2) using end-user generated data (customer satisfaction
classification based on web usage metrics). The results of the customer satis-
faction score formula somewhat correspond with the results of the web usage
metric classification (91.2% accuracy).

The specific contributions of this thesis with regards to the vicious circle that high-
lights interdependent key drivers impeding the productization of analyses are:

- The technical, organizational, and people-related challenges in operationalizing
analyses were investigated in an interview study and a survey. In addition to
that, the study contains views of two different angles: users of analyses and
providers of analyses. As a result, the presented challenges are not only based
on multiple data sources but also on different perspectives, thereby providing
an overall view of the problem.

- The derived challenges form the following interdependent key drivers impeding
the operationalization of analyses: 1) Lack of priority, time, and resources;
2) low data quality; 3) inability to cross cultural gap; 4) ineffective prototypical
analysis; and 5) inability to prove value. By being aware of the challenges and
the key drivers, practitioners can take a more targeted approach at breaking the
circle by specifically focusing and tackling the points that they can influence.

The specific contributions of this thesis with regards to the framework for the pro-
ductive use of machine learning in software analytics and business intelligence and
the adoption of a data-driven mindset in B2B software-intensive companies are:

- A literature review on data management and processing, model building, and
model deployment was conducted and the key activities in each of the stages
were extracted. Based on this, an end-to-end framework is presented that com-
prises all activities from data collection to retraining of deployed ML models.

- The framework consists of three iterative cycles: prototyping, deployment, and
update cycle. By outlining the transition between the cycles, the framework
guides practitioners in getting from a prototypical analysis to an automated
and deployed analysis. Especially since the evaluation of the framework indi-
cates that the separation of activities across the conceptual cycles creates the
perception that the overall, potentially overwhelming, process now consists of
several smaller steps that are easier to handle.
• The results of the validation indicate that the activities of the framework are consistent with the activities performed in practice. In addition to that, it is a practical tool to keep an overview of all required steps and to efficiently define and plan upcoming activities.

• Stakeholders go through different phases during the process of adopting a data-driven mindset. The phases identified in Chapter 9 are: skepticism, interest, exploration/experimentation, and collaboration. Depending on the stakeholders’ phase, data engineers and data scientists need to act and react differently according to the stakeholders’ needs. By being aware of the different phases and by knowing what type of interaction, communication, and collaboration their stakeholders currently need, practitioners can adapt their activities accordingly in order to drive the adoption of a data-driven mindset.

10.5 Further Research

This section summarizes areas for future research building on the work that is presented in this thesis. Chapter 3 presents an approach for incrementally generating high-level insights from low-level metrics for product management stakeholders. Chapter 5 builds on this approach by extending it to other types of stakeholders. Key observations were the relevance of different data sources to stakeholders and the reusability of components. Future work could be invested in further examining the concept of primary data sources and the relationship between data sources and different stakeholder types.

Chapter 4 presents an approach for predicting customer churn in B2B contexts. After the study was completed for the selected case, the same approach was implemented for two additional B2B products. Therefore, future research could be dedicated to further validating the approach by comparing the results of all three case products. The model presented in Chapter 6 is based on the assumption that there exists a direct relationship between customer satisfaction and web usage metrics, which was indicated in existing literature. However, a direct validation through customers was not possible due to company restrictions. Therefore, upcoming research could be invested in examining the relationship between web usage metrics and customer satisfaction more closely and in further validating the approach by comparing it with real customer feedback.

Moreover, the solutions presented in Chapter 7 are validated by a group of experts. Therefore, the validation of the study is limited to their experience and perceptions. A long-term evaluation of applying the proposed solutions in practice could be conducted in future research.

The same applies to the end-to-end framework for a productive use of ML presented in Chapter 8 and the model for driving the adoption of a data-driven mindset in B2B software-intensive companies presented in Chapter 9. Both approaches are validated based on short-term observational data. A long-term validation covering the full ML cycle would allow to validate the impact and evaluate potential bene-
fits of establishing a data-driven mindset and its effects on customer satisfaction or company success.
Bibliography


245


252


From operational data to business insights: Adopting data-driven practices in B2B software-intensive companies

Modern software engineering practices such as DevOps enable product teams to shorten their release cycles by developing, deploying, and operating their products in a continuous loop. Along the way, vast amounts of data are generated throughout the entire product lifecycle from development to deployment to operations. This type of data can play a key role in continuously monitoring and improving products and processes in a data-driven way. The goal of data-driven software engineering is to no longer take decisions based on opinions and gut feelings, instead to rely on and retrieve insights from product and customer data in order to take informed and evidence-based decisions. However, the adoption of data-driven practices within organizations is difficult as it requires a major change of mindset and a large amount of trust and belief in the value of working in a data-driven way. Oftentimes, data scientists and data engineers are becoming part of traditional software engineering teams. Their diverging ways of working and mindsets towards data pose a risk for emerging conflicts between the two roles.

Moreover, the quality and usefulness of an analysis and its results highly depend on the availability and quality of data. Precisely this causes uncertainty in applying data-driven approaches in business-to-business (B2B) contexts. The volume of data being generated or even available in B2B contexts is often significantly lower compared to business-to-customer (B2C) domains. For one, this is caused by a more limited number of customers in B2B contexts. For another, the customer of a software-intensive product is often not the end-user of the product but rather an intermediate entity. As a result, data generated by the end-user is often more complex to analyze or not accessible at all. These limitations often lead to a lack of confidence in the data which in turn makes it difficult to drive the adoption of data-driven practices in B2B software-intensive companies.

This thesis investigates the adoption of data-driven practices in multiple, real-world B2B product teams. The studies are based on case study research conducted across three case companies and twelve product teams. Both qualitative data resulting from interviews and long-term collaboration and quantitative data in the form of a survey were collected. The thesis is structured in three parts addressing three overarching research questions.

First, Chapter 2 presents the current state of how product teams in distributed organizations communicate with each other, what kind of information, feedback or data they share, and how they are making and communicating decisions. The results of the case study highlight a number of conflicting interests between different actors which lead to different types of coordination challenges of teams in B2B software-intensive companies. For example, there is a lack of established processes to collect and analyze data which makes it difficult to obtain a broad picture of all customers.

Based on this challenge, the second part of the thesis comprises four studies that introduce conceptual models for generating technical and business insights for a variety of roles (e.g. software architects, product managers, or sales) based on operational
data. Chapter 3 and Chapter 5 investigate the information needs of various stakeholders involved in the development and distribution of B2B software-intensive products. Based on their needs, operational (customer and product) data is being analyzed to generate stakeholder-targeted information for data-driven decision making. Key aspects of the models are iterative feedback and specification cycles as well as the use of analysis outputs as inputs in subsequent levels and the reuse of appropriate components across stakeholders. Chapter 4 and Chapter 6 specifically focus on how to deal with B2B-specific data characteristics for analyzing customer churn and customer satisfaction in B2B contexts. In both cases, end-user generated data is systematically mapped to customer-generated data.

The third part presents technical, organizational, and people-related challenges in productizing (automating and deploying) prototypical analyses generating insights in the context of data-driven software engineering. A subset of these challenges is further investigated in two additional studies in order to identify and develop approaches for addressing the challenges. Chapter 7 identifies the challenges of operationalizing AI-based analytics for use in practice. The results of the case study indicate that there are inter-related key drivers that form a vicious circle impeding the productization of analyses. Chapter 8 and Chapter 9 both build on the challenges identified in the previous study. Chapter 8 describes an end-to-end process for implementing and deploying ML-based software analytics and business intelligence solutions and Chapter 9 investigates how to drive the adoption of a data-driven mindset in B2B software-intensive companies.

Finally, Chapter 10 outlines how each research question is addressed by the studies presented in this thesis. It also summarizes the threats to validity and key contributions of the study, before concluding with a general outlook on further research.
Appendix B: Curriculum Vitae

Iris Fgalist is a platform and data architect in the corporate technology department at Siemens AG. She has a background of six years in software development and data analytics, where her work primarily revolves around the application of data-driven software engineering practices in B2B software-intensive companies. In 2018 she received a MSc degree from the Ludwig Maximilian University in Munich, Germany.