619 Degree Project in Management
Stockholm School of Economics
Ludvig Svanström 24088 & Tor Alm 24089
May 2020

Managers putting Turing's ideas to the test

A critical success factor study about machine learning projects in software organizations

Abstract

Through a qualitative multiple-case study in a comparative design, this thesis aims to discover the critical success factors for implementing machine learning (ML) for software companies. The empirical material consists of interviews and documents from four case organizations of both failed and successful projects. A theoretical framework based on the school of project management called critical success factor research is used to analyze the findings. The findings illustrate six critical success factors for ML implementations in software companies. Clear objectives and goals, Effective project management methodologies, and Realistic schedule have been found as success factors in both the software implementation literature and this study. Three new factors, Experimentation over planning, Deep understanding of the dataset, and Solution over technology orientation, have been found in this study. These differences indicate what project managers need to master in order to implement ML models in software organizations successfully. The study also increased the understanding of project success factors into a new context of the developing subfield of ML implementation.

Keywords: Critical Success Factors, Machine Learning implementation, Software Implementation, Project Management, Software company, AI, ML

Thesis supervisor: Adis Murtic, Visiting Researcher, *Department of Entrepreneurship, Innovation and Technology*

Examiner: Laurence Romani, Associate Professor, *Department of Management and Organization*

Authors & Student IDs:
Ludvig Svanström 24088
Tor Alm 24089

Number of words: 11,976

Acknowledgements

The authors of this study would like to extend their gratitude to all helpful individuals for their trust and assistance in writing this thesis:

- Adis Murtic, for your engagement and guidance
- Patrik Tran, for insightful discussions on machine learning
- All members of our supervisory group, for your feedback
- All the interviewees who took time from their valuable working hours to share their deep practical knowledge around the exciting phenomena of getting machine learning into production.

Thank you,
Ludvig & Tor

Table of Contents

Definitions

Artificial Intelligence (AI)	The study of agents that receive precepts from the environment and perform actions (Russel & Norvig, 2010 p.8).
Machine Learning (ML)	A branch of artificial intelligence that systematically applies algorithms to synthesize the underlying relationships among data and information (Awad & Khanna, 2015 p.1).
System	A set of connected things or devices that operate together (Cambridge Dictionary, 2019a).
Algorithm	A list of instructions for solving a problem (Cambridge Dictionary, 2019b).
Critical Success Factor (CSF)	The limited number of areas in which results, if they are satisfactory, will ensure competitive performance (Rockart, 1979).
Information Technology (IT)	All forms of technology used to create, store, exchange, and use information (Moraveck, 2013 p.2).
Information Systems (IS)	A combination of technology, people, and processes that organizations use to produce and manage the information (Moraveck, 2013 p.2).
Application Program Interface (API)	A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service (Lexico, 2020a).
Central Processing Unit (CPU)	The part of a computer in which operations are controlled and executed (Lexico, 2020b).
Graphical Processing Unit (GPU)	A specialized hardware component that were originally developed for graphics applications (Goodfellow et al., 2016 p.444).

Table 1. Definitions of used abbreviations and terms

1. Introduction

1.1 Background

Artificial intelligence (AI) was suggested by Alan Turing in a landmark paper from 1950 as a theory of machines being able to act intelligently (Russel & Norvig, 2010 p.2). Although the paper was speculative at the time, recent advances in computational power, mathematical models, as well as the expanding quantity of available data have enkindled managers' trust in betting on this technology to solve their problems (Ransbotham et al., 2017).

In the last decade, experts have seen radical advancements in the performance of a specific branch of AI called Machine learning (ML). Time and again in computer science academia, ML has proven to be the most successful technology to solve complex computational problems such as analyzing and understanding images, videos, and languages (Jones, 2014; Devlin et al., 2019). Hence, allowing computers to act more intelligent as Turing once proposed.

85% of executives across a broad range of industries think that AI will provide their organization with, or sustain, a competitive advantage (Ransbotham et al., 2017). Regarding ML specifically, some of the world's most valuable companies are making substantial investments into the technology and are integrating it into their core products with commercial success (Fortune, 2020). Examples include Amazon's cashier-less stores, Tesla's self-driving vehicles, and Google's voice assistants (Inside Big Data, 2018; Financial Times, 2019; Forbes, 2019).

Despite the promises of AI, many organizations are not successful in their implementation efforts (Hosanagar & Saxena, 2017; Fountaine et al., 2019). Practical implementations of the technology are found to be challenging, as only 5% has extensively incorporated AI into their offerings and processes (Ransbotham et al., 2017).

There are some pioneers in the field of successful ML projects. The information and communication technology (ICT) sector is frequently cited as the most digitally ready industry, with the potential to pick up advances in AI (Gandhi et al., 2016). Software companies form a crucial part of this industry, as they provide the products and services within the field (Vinnova, 2013). Therefore, in accordance with Koc's (2007) argumentation, these companies already have expertise in the highly complex task of creating software, which might be transferable to ML tasks.

ML requires digital data to act intelligently (Goodfellow et al., 2016 p.2). Digital companies, with an online software product, have access to significantly more data than traditional businesses (McAfee & Brynjolfsson, 2012).

The combination of access to relevant skills as well as access to large quantities of data makes software companies especially suited for taking advantage of ML techniques.

The authors argue that the strand of research most suitable to further our understanding of why implementation efforts are unsuccessful is called Critical Success Factor (CSF) research. This foundational field within project management uncovers the critical factors of projects that are essential to its success (Müller & Jugdev, 2012). Hence, the academic field of CSF research could be applied to assist practitioners in furthering the understanding of the characteristics of successful ML projects.

1.2 Earlier studies and research gaps

The authors of this study argue that there is a research gap formed around CSFs in research for ML implementation projects primarily for three reasons.

Firstly, research into how organizations should strategize and implement AI has been conducted but is primarily focused on an organizational level (Brock & von Wangenheim, 2019; Fountaine et al., 2019; Raj et al., 2019; Polyzotis et al., 2017).

Secondly, ML implementations are related to the field of software implementations, which is a well-researched area for CSFs on a group level. There are distinct differences between ML system development and regular software development, which potentially implies critical differences in implementation successes as well (Sculley, 2015; Arpteg et al., 2018). This is not the first development in the field that might render older CSFs to be obsolete. In the 1990-2000s, CSF research required an updated focus from waterfall oriented mainframe projects to smaller agile projects (Schmidt et al., 2001). Therefore, the authors of this thesis argue that ML potentially provides the basis for another shift in the research field of software implementation CSFs.

Lastly, the authors argue that AI is too big of an umbrella term for a broad set of technologies to find generalizable conclusions. In line with authors, such as Goodfellow et al. (2016 p.8), who argue that ML is the only viable approach to building AI systems that can operate in complicated, real-world environments, this paper distinctly focuses on the specific AI technology ML.

1.3 Purpose and research question

Due to the promises of ML, the shortcomings of the implementations, and to close the apparent gap of current studies, this study's research question is:

What are the critical success factors for implementing machine learning for software companies?

Definitions

A *critical success factor* is defined by Rockart (1979) as *"the limited number of areas in which results if they are satisfactory, will ensure competitive performance."* According to Rockart (1979), if these factors are not adequately addressed, performance will not be satisfactory.

A *software company* in this study is a firm that develops software as their main product or service and generate revenues by granting access for users of this software, following the definition provided by Vinnova (2013).

1.4 Clarification & delimitation

In this study, the focus will be upon the implementation and project leading of ML projects. The scope of the research, therefore, makes an implicit assumption that the choice of this technology most efficiently solves the company's business problem. This acknowledges that the topic of how businesses come to choose this specific technology in the first place also needs further study. However, as use cases are starting to reveal themselves, this study aims at examining what the authors perceive to be the most critical barrier to adoption - the implementation.

The issue of evaluating whether a project is to be considered a success or not is an active debate (Remus & Wiener, 2010). ML projects, in general, are hard to evaluate as they can be challenging to maintain and service in the long term (Arpteg et al., 2018). This thesis does not aim to contribute to the extensive literature on project success criteria. Instead, the evaluation will be based on formal, unambiguous measures. In order to have a comprehensive definition, success is defined if the output is used, and the project fulfilled the time, cost, and performance constraints set up in the early phases of the project, as these are recommended for evaluation by, for example, Pinto & Slevin (1988).

2. Theoretical framework

This section describes and explores the theoretical framework based on project management and CSF research. The section concludes by discussing why ML might differ from previous research and discusses the criticisms of the chosen theoretical framework.

2.1 Usage of theory

As the topic of the thesis aim to discover the CSFs of ML implementation efforts, it is a multidisciplinary undertaking in the areas of project management, software development, and data science. However, this thesis aims to contribute to the broad field of management in the context of project management. For an audience not familiar with the term AI and its subdomains, an additional in-depth introduction is included in Appendix 9.1.

The literature review provides an overview of the fundamentals of the project life-cycle. In addition, an overview of the topics within CSF research is provided, as this method is the most common way of analyzing the contributing factors of project success (White & Fortune, 2006). The project success school, which CSF research constitutes one of two categories in, form one of 9 central schools of project management research (Bredillet, 2008). Others, such as Söderlund (2011), have found that the Factor school of CSFs form one of seven major project management schools, which signifies its importance in the field and it's potential in assisting the researchers in answering the research question.

The review concludes by discussing the potential differences of ML implementation projects and regular software development projects, which calls for further investigation into ML specific CSF research.

2.2 Project management

The Oxford University Press defines a project as "An individual or collaborative enterprise that is carefully planned to achieve a particular aim" (Lexico, 2020c). According to Tonnquist (2016 p.8-9), a project is suitable as a working method, when there is a need for coordination inside an organization, or to pool collective resources from multiple organizations.

Often, in order to describe projects, they are subdivided into different distinct phases. Heagney (2011 p.21) argues that there are many different ways of dividing a project into phases. The most general way of division contains four phases: Idea & pre-study, Planning, Execution, and Finalization (Tonnquist, 2016 p.17-18). These are elaborated upon below to familiarize the reader of the different activities associated with the different phases of the project.

2.2.1 The different phases of a project

Idea & pre-study

In the pre-study, the idea is developed into a business case, and the prerequisites to finalize the project are evaluated. In this phase, the scope of the project is established by determining the objectives and goals. Then, requirements and specifications are defined (Tonnquist, 2016 p.44-64). Pinto (2016 p.13), who calls this phase “the Conceptualization phase”, adds that stakeholders and the necessary resources are identified as well.

Planning

In the planning phase of the project, the road toward the project goal is specified (Ibid, 2016 p.13). Commonly, the time plan is determined and detailed activities, or agile sprints, are scheduled. The needed resources will be defined alongside the schedule, and risks need to be handled (Tonnquist, 2016 p.128-130).

Execution

The execution phase consumes most of the resources of a project. In this phase, it is essential to keep track of the progress, handle changes, and update plans. In most projects, the implementation and delivery of the project results lie within the execution phase (Tonnquist, 2016 p.224-225).

It is in the execution phase that the project management methodologies are mostly applied. Agile methods are especially popular within software development, but the method is spreading to other sectors as well (Pinto, 2016 p.369-376). Agile methodology is most suitable when the projects need to deliver usable output quickly, it has unclear requirements, and the end product is hard to visualize (Tonnquist, 2016 p.39). In short, Agile methodology entail working in short development cycles, so-called sprints, with delivery and feedback often. The responsibility of what to prioritize, plan, and deliver upon rests with the members of the team. Agile teams also strive to visualize ongoing work by, for example, using Kanban-boards (Tonnquist, 2016 p.32-35; Gustavsson, 2019 p.131-154).

Finalization

When the result is delivered, the finalization phase entails evaluating the project and closing the project team. Often, lessons learned are produced to support reflection among project members (Tonnquist, 2016 p.321-345).

Beyond understanding the fundamental elements of projects and essential tools to execute one, a significant portion of research has gone into the understanding of how to evaluate and enhance projects, as explained below.

2.2.2 Research into project success

At the core of project management research is the ongoing topic of project success, which started in the sixties and continues to be relevant to this day (Cooke-Davies, 2002; Müller & Jugdev, 2012). Cooke-Davies (2002) outlines the vital difference between the two concepts that make up the school of project success research, the terms *project success factors* and *project success criteria. Success factors* are inputs that directly or indirectly lead to the success of a project, while *success criteria*, in turn, are the measures by which the outcome of the project should be measured to determine its success (Cooke-Davies, 2002). The research on CSFs has several strengths in software project management. First, when asked to articulate CSFs or reflect on their practices, managers can hone their understanding. Secondly, the method can provide understandable, relevant, and useful information for practitioners (Boynton & Zmud, 1984; Henderson et al., 1987).

Closely related to the study of CSFs is the tangent of research into the study of “Pitfalls” or “Critical Failure Factors” (CFFs), i.e., problems with project execution (Belassi & Tukel, 1996; Yeo, 2002; Whittaker, 1999; Boehm, 1991). This study focuses on the body of CSF literature and does not make a distinction between CSFs generated by studying failures or successful examples, which is in line with other researchers such as Fortune & White (2006) and Belassi & Tukel (1996).

2.2.3 CSFs for project management

The study of success factors is well developed, and many studies have been investigating the CSFs for project management in general (Fortune & White, 2006). Different industries and contexts tend to have their separate factors contributing to project success (Lim & Mohamed, 1999). In order to build, train, and deploy ML models, software is needed (Marsland, 2014 p.11-21). Hence, the literature review pivots into the success factors specific to the field of software development projects. This is a significant narrowing of the scope, as software development projects have significantly different characteristics, even to other engineering projects (Fairley, 2009; Jain, 2008).

2.2.4 Success factors within the software development context

Software projects tend to have a high degree of complexity, which complicates project management in this context (Reel, 1999). In this study, the definition of a software development project includes the terms “IS projects” and “IT projects”. Even though IT is a subset of IS, both are included as the classifications in reality, even in many research articles about the topic, often are used as synonyms (Moraveck, 2013 p. 3). The success rate of software development projects has increased significantly over the last 20 years. Partly, due to an increased understanding of the project management playbook and, partly, due to a broadened understanding of what constitutes project success (PMI, 2017).

Software development projects are, however, not merely one type of project, but a multi-faceted collection of projects. There have been several CSF studies within these specific contexts (Ramaprasad & Williams, 1998). For example, the CSF approach has been applied to software development projects in general (e.g., Reel, 1999; Yap et al., 1992; Poon & Wagner, 2001; Karlsen et al., 2005) and for more specific IS projects such as Enterprise Resource Planning (ERP) systems implementations (e.g., Nah et al., 2003; Ghosh & Skibniewski, 2010) or Customer Relationship Management (CRM) systems implementations (e.g., Rahimi, 2009; Croteau & Li, 2003). Furthermore, there have been some studies examining IS implementations in the public sector using the CSF approach as well (e.g., Ahmed et al., 2018; Carlton, 2017).

This plethora of research has led to many different CSFs ranging from clear objectives and goals (Keil et al., 2002; Schmidt et al., 2001; Reel, 1999), support from top management (Pinto & Slevin, 1989; Sofian, 2003; Kamal, 2006), realistic and good scheduling (Humphrey, 2005; Taylor, 2006), project understanding (Baccarini & Collins, 2003), good planning (Frese & Sauter, 2003), competent team members (Somers & Nelson, 2001; Wiener, 2006), clear and frozen requirements (Kappelman et al., 2006), client/customer involvement (Sauer & Cuthberson, 2003; Standing et al., 2006; Charette, 2005) and several more.

Nasir & Sahibuddin (2011) performed a comparative review of 43 articles of CSFs in software projects. The result of the review was 26 CSFs. The top five occurred in more than 50% of the reviewed papers. These top five CSFs are Clear Requirements and Specifications, Clear Objectives and Goals, Realistic Schedule, Effective Project Management skills and methodologies, Support from top management. Within software development projects, project leaders are recommended to be attentive to these five factors (Nasir & Sahibuddin, 2011).

2.2.5 Why CSFs for machine learning projects could be different

There are, however, several differences between regular software implementation and ML-based solutions. For one, how the model derives its output is more difficult to explain. Furthermore, the importance of system learning, as well as acquiring big datasets, are other vital differentiators. ML implementations also tend to have a more iterative trial and error approach to developments (Prem, 2019).

In software engineering, these differences between ML systems and non-ML systems are becoming more visible and has started to be addressed. Instead of hard coding rules, ML systems acquire them by observing data directly. This calls for data to be scrutinized and tested in the same way that code is tested today. However, currently, tools and best practices are still lacking (Arpteg et al., 2018). This emphasis on data, and the need for data preprocessing, is a significant part of ML development, and adds new dependencies in the projects (Zhou et al., 2017; Bengio et al., 2013).

Furthermore, there are not just differences in the development phase. According to Sculley et al. (2015), there are several issues related to the maintenance of ML systems that needs to be addressed. This is because ML systems have all the maintenance issues as regular code and some ML specific issues. Developers of ML systems need to think about these additional maintenance issues in order to secure the long term success of their system (Sculley et al., 2015).

All these differences, be it in the collection of data, the number of iterations, the effort estimations of the projects, or the maintenance issues with ML systems, can create significantly altered tasks compared to a regular software development project. This difference in the workflow at the practical and technical level might be reflective of the need for a changed set of behaviors and success factors on the level of the project manager as well.

2.2.6 Criticism against the use of CSFs

There is an ongoing debate around the validity and usefulness around using CSFs, and two main criticisms towards the use of CSFs can be identified in the literature (Fortune & White, 2006). The first is that interrelations between factors tend to become overlooked (Nandhakumar, 1996). Secondly, critics argue that CSFs disregard the dynamics of project management, as different factors can have varying importance over time as the project progresses (Larsen & Myers, 1999).

This bachelor thesis does not aim to add to this debate. Instead, the authors argue in line with Kieser & Nicolai (2005), who argue, that CSF research can bridge the gap between academia and practice. In line with many other authors, (e.g., Alhassan et al., 2019; Yeoh & Koronios, 2010), the authors of this thesis argue that CSF research continues to be an exciting and valid pursuit.

2.3 Theoretical discussion

The pursuit of project success research is a well-developed subfield within the broader field of Project Management. CSF research has been developed over the years, and it continues to be relevant due to its connections between academia and practice. Despite the subfield's relative maturity, there seems to be no broad consensus among researchers around a limited set of CSFs in the context of software projects. Although some researchers have broadly summarized the patterns within the field (Nasir & Sahibuddin, 2011). Potentially, this can be explained by project management's inherently dynamic context, which makes different CSFs important in different phases of projects, or the broad range of software projects included in the literature review.

Finally, the different nature of ML projects calls for further research into this area, as CSFs in this context might differ from regular software projects. Alternatively, perhaps only a subset of the CSFs mentioned in the software project literature is relevant for ML projects.

This makes it imperative for both researchers and practitioners to develop their understanding of what makes ML projects successful. Therefore, this study aims to explore which CSFs exist within ML implementation projects.

3. Method

In this section, the researchers' methodological assumptions are clarified. In addition, the research design is described along with the sampling approach and ethical considerations. The section concludes with an introduction to the case companies.

3.1 Choice of method

3.1.1 A study based on objectivism and positivism

AI, among other digital technologies, is expected to be an intensive disruptive force for individuals, organizations, markets, and society at large (Puaschunder, 2019; Thirgood, 2017; Haenlein, 2019). In a dynamic context, such as the current AI development, positivist qualitative research allows researchers to explore and describe phenomena despite its ambiguous nature as argued by Su (2018). Hence, in this study, a positivist stance was taken, in line with the argumentation of Burrell & Morgan (1979 p.5), to explain events in the social world by discovering regularities and causal relationships between its constituent elements. This nomothetic approach entails that the goal of the research is to both explain and predict what happens in the social world, as argued by King et al. (2019 p.12). The above-specified rationale for choosing a positivist qualitative method would classify this study, according to King et al. (2019 p.20), as "qualitative neo-positivist".

In conjunction with the positivist stance, it follows that an ontological position of objectivism is assumed. This implies the assumption of an objective and external reality. The view of the researchers is that the external reality is apprehendable and, therefore, summarizable as well as generalizable in line with the description by Bell et al. (2019 p.26).

3.1.2 An abductive qualitative study

Theory and empirics have been studied and collected simultaneously through the research process in an abductive manner. The approach of the continuous exchange between the theory and empirics was chosen as it yields a deeper holistic understanding of the phenomena, in accordance with the reasoning of Bell et al. (2019 p.24). Furthermore, an abductive stance decreases the tendency to merely approve existing knowledge (Alvesson & Kärreman, 2007). As the authors suggest that there might be differences between earlier CSF research on software implementations and ML implementations, relying solely on deductive or inductive reasoning would not have been as beneficial.

This research is conducted through a qualitative inquiry. The choice of a qualitative method was built upon the ontological stance, in line with Cassell's et al. (2017 p.18) reasoning, that there is an external reality that can be summarized and understood, but not easily quantified. The positivistic qualitative method aims to acquire knowledge through non-statistical means, regularities, and causal relationships between different elements of reality (Su, 2018). This resulted in the choice of a qualitative methodology as the contextual character of qualitative research is more suitable than quantitative research due to its more holistic depiction of a phenomenon, as argued by Rynes & Gephart (2004 p.455).

3.1.3 Multiple case study approach

The exploratory study was based on multiple case studies in a comparative design. Bell et al. (2019 p.68-70) describe the comparative design as an extension of the case study in qualitative research. Multiple case studies are chosen as a methodical approach to find generalizable findings and best practices (Yin, 2014 p.62; Brown & Eisenhardt 1997; Su, 2013). This method suited the research as the aim was to find success factors for the implementation of ML to answer the research question. Furthermore, as the research regard the contemporary phenomenon of ML implementations, the case study format was considered especially suitable, as argued by Yin (2014 p.12). As multiple cases allow for replication, due to cases being seen as experiments, this approach tends to be stronger and more robust than single-case study research (Herriott & Firestone, 1983).

In any case study, it is important to define the unit of analysis in order to connect the research to the current body of literature (Bell et al., 2019 p.69-71). Furthermore, by bounding (i.e., clarifying) the case topic, it can be distinguished from the context (Yin, 2014 p.33). The implementation efforts studied in this thesis was performed on a group level, which also forms the basis of our unit of analysis.

The criticism against multiple case studies that contrasts between cases are accentuated at the expense of context, raised by Dyer & Wilkins (1991) is valid. However, the difficulty of balancing insightful details about the context and what Lofland et al. (1995) called 'descriptive excess' can be viewed as a constant challenge for all qualitative research (Bell et al., 2019 p.368). This thesis follows the argumentation of Bell et al. (2019 p.69-70), who argue that multiple-cases can improve theory building. Furthermore, they can help researchers find interesting contrasting characteristics of the cases, which later can act as an enabler for theoretical reflections about these findings (Bell et al., 2019 p.70).

3.2 Choice of cases

3.2.1 Software organizations

In case studies, in contrast to a regular sampling logic, individual cases should be seen as experiments and not respondents to a survey, as Yin (2009 p.37-39) has argued. Therefore, case studies in a replication logic are an empirical investigation that allows for analytic generalization (Mills et al., 2010 p.21-22). To that end, cases were strategically and purposively selected. With a purposive selection methodology, the best strategy depends on the research context and nature (Bell et al., 2019 p.389-391; Palys, 2008). This strategic selection is advantageous to this study as it allows the researchers to select cases specifically tied to the end objective of the research.

The screening for cases was made by both searching the internet, looking through media and trade reports, and talking to an industry expert. The criteria for being considered in the study was that the case should have been done within a software company, as well as that the project should have been finished. This screening yielded over a dozen identified candidates. From the screened candidates, four were selected due to their diversity of characteristics and fit with the research replication design.

In order to validate the research methodology a small pilot interview was conducted. The pilot interview was conducted with an industry expert and resulted in some revisions to the interview guide and research focus.

3.2.2 Interviewees

In accordance with the reasoning of Bell et al. (2019 p.390), participants were selected based on who had the potential to inform the research questions and enhance understanding of the phenomenon under study. Hence, interviewees in the cases were chosen based on their role as project leaders, and therefore high involvement in respective projects, which the authors argue ensures comparability between cases and is aligned with the group-level analysis of this research. All the included cases had one person equipping this role, which simplified interviewee selection. When possible, other participants in the projects were interviewed as well to get more perspectives.

Three of the seven interviews were conducted face-to-face, and the other four was conducted via video interview. King et al. (2019 p.120-121) mention some issues such as lousy broadband connection and difficulties establishing rapport when using telephone or video interviewing. However, this study had limited choice due to the ongoing global pandemic, entailing global social distancing measures, which occurred at the time of this study being conducted. This was addressed by having an initial conversation through email with the interviewee, both for clarification and scheduling purposes, which enabled rapport to be established in line with what Deakin & Wakefield (2014) have argued. The researchers would argue that the video interviews offered a face-to-face experience while also retaining privacy and flexibility, as discussed by Hanna (2012).

3.3 Interview process

3.3.1 Collection of empirics

The empirics collected in the research consist of both interview data, documents, field notes, and publications from the companies. This ensures the triangulation of knowledge and can lead to more robust findings (Yin, 2014 p.119). All documents were assessed through the four criteria of authenticity, credibility, representativeness, and meaning in line with Scott's (1990) classification.

According to Yin (2014 p.110), interviewing for case research usually resembles guided conversations or unstructured interviews. In this study, semi-structured interviewing was used for two reasons. First, as the study involved multiple cases, the structure was needed to enhance the reliability and validity, as argued by King et al. (2019 p.20). Secondly, the semi-structured interviews provided the researchers with enough flexibility while maintaining comparability, as has been argued by Bell et al. (2019 p.436-438), which was considered important to preserve validity and reliability.

The interviews were recorded and transcribed word for word. To ensure the quality of the transcriptions, an often overlooked threat to qualitative research, according to King et al. (2019 p.196-200), the researchers cross-checked all transcriptions. No interviewee declined the use of a recording device.

The interviews were supported by an interviewer guide that was developed with the literature review in mind [Appendix 9.2]. However, to avoid potential preconceptions about the subject, the interview guide included broader questions as well. The interviews ranged for approximately one hour to one-and-a-half hours.

3.3.2 Processing and analysis of empirics

As the researchers personally transcribed the interviews, one could argue that this was the first step of the analysis, in line with Langdridge & Hagger-Johnsson (2009). The analysis followed a general thematic approach following the process outlined in Bell et al. (2019 p.530-531). A thematic analysis suited the research due to its flexibility and ability to extract themes related to the research question in line with Bell's et al. (2019 p.519-520) argumentation.

Codes were continuously and iteratively developed alongside comparisons with the literature and the researchers prior understanding in an abductive manner. The codes was developed after the material had been transcribed to find similarities, keywords, and topics in the interviews and through the complementary material. Themes were developed in line with Ryan & Bernard's (2003) recommendations by examining the initial codes, for example, looking at repetitions, similarities, and differences between cases, as well as missing data. The themes were also validated through triangulation by looking at supporting material such as websites, articles, and notes.

In the Empirics section, the themes considered revealing genuinely critical project success factors were discussed. The empirics revealed that the following six CSFs emerged from the respective project phases:

CSF in ML	**Project Phase**
Clear objectives and goals	Idea & pre-study
Experimentation over planning	Idea & pre-study
Realistic schedule	Planning
Deep understanding of the dataset	Execution
Effective project management methodologies	Execution
Solution over technology orientation	Execution

Table 2. The CSFs emerging from the themes in respective project phase

Then, the themes revealing critical factors were interpreted and analyzed with the literature. The theoretical frame allowed for comparisons and a deeper understanding of the similarities and differences between ML and software implementations.

3.4 Ethical approach and implications

The four ethical principles that Diener and Crandall (1978) have outlined has guided this thesis. The goal of the thesis was to enhance project managers' ability to implement ML in their companies to minimize the waste of resources and involved no vulnerable persons, which ensured avoidance of harm to participants.

The participants and case organizations' right to privacy have been prioritized above the disadvantages of anonymization of case studies outlined by Mills et al. (2010 p. 26). The researchers consider transgressions of this ethical principle unacceptable as Bell's et al. (2019 p.123) have argued. Furthermore, the researchers are sympathetic to the anonymization, due to the business criticality of the information contained in the cases, and the participants' willingness to discuss organizational failure.

3.5 Critique of the choice of method

There are disagreements regarding the use of the terms validity and reliability in the context of case study research (Bell et al., 2019 p.65). In this thesis, Yin’s (2014 p.46-49) classification of construct validity, external validity, and reliability will be used.

3.5.1 Construct validity

Construct validity concerns finding measurements that correspond to the concepts being studied. The difficulty of developing relevant operational measures as to avoid biased judgments is well known in case research (Yin, 2014 p.46).

As multiple sources of evidence have been used, this strengthens the researchers' confidence in a satisfactory level of construct validity. As research participants have been given a chance to review the case description draft, i.e., what King et al. (2019 p.261) calls “member validation”, this minimizes the risk of misunderstandings and misrepresentations. This ensured that the cases fulfilled the participants' requirements of anonymity and also provided a chance for clarifications.

3.5.2 External validity

The issue of generalizability is an extensively discussed topic in case research (Bell et al., 2019 p.64-65). Lee et al. (2007) argue that generalizability is not a strength of the case study method. Others, such as Flyvbjerg (2006) and Yin (2014 p.48), disagree. Nevertheless, most academics would argue that purposively selecting cases, as has been done in this study, implies that the findings are not fully generalizable due to the approach as a non-probability sample, much like the majority of qualitative studies in general (Bell et al., 2019 p.389). However, rather than considering cases as samples, one should see it more as an experiment (Yin, 2014 p.48). This study was based on multiple cases, and a replication logic in line with the Yin-Eisenhardt approach has been followed, which strengthens the external validity of the research (Mills et al., 2010). By strategically selecting the cases, in line with Eisenhardt's (1989) influential article, this thesis can provide robust findings that later studies can refute or validate by further experimentation.

3.5.3 Reliability

When discussing the reliability of case studies, Yin (2014 p.49) argues that one should reach the same conclusion if one were to do the same case again, as opposed to a new one. As the access to cases can be dependent on personal connections, it might be near impossible to replicate this case study exactly. Instead, the authors of this thesis strived to fulfill what Yin (2014 p.49) argues, is the goal of reliability in case studies: minimize errors and biases in the study.

3.6 The researchers' access to participants

Access to the screened organizations was available primarily through the industry expert. Both researchers argue that due to the absence of financial or other incentives, the industry expert, as well as the interviewee subjects, had no reason to affect the end result of the study adversely.

3.7 Introduction to the case organizations and projects

The following table provides an overview of the projects studied in this thesis. For the more detailed case descriptions, please see Appendix 9.3.

Case (1-4)	**Organization**	**Number of employees**	**Nationality**	**Successful ML implementation**
Fishing forecast prediction	Social media platform with the help from Modulai, an ML consultancy firm	100-200	Swedish	Yes
Spam picture classification	E-commerce platform	Above 1000	Non-European	No and Yes*
Customer retention predictor	Media streaming platform	Above 1000	European	No
Language pronunciation classification	Educational technology platform	Less than 50	European	Yes

Table 3. Case description of participating organizations

*Case 2 involves two separate projects, as the project was attempted twice where the first failed (Case 2a), and the latter succeeded (Case 2b).

4. Empirics

In this section, the themes and findings that emerged from all four cases are presented. The section is structured according to the four project phases Idea & pre-study, Planning, Execution, and Finalization, to illustrate the different findings for each specific phase.

4.1 Idea & pre-study

In the project phase Idea & pre-study, two distinct themes each reveal a CSF. The themes are Setting objectives and goals, and Setting up requirements.

4.1.1 Setting objectives and goals

A discovered commonality among the successful cases was their shared initial clarity of objectives and goals. That is, they had both unambiguous goals with clear key performance indicators. Case 3 focused on trying to predict retention rate, and Case 2 focused on how to increase spam-detection. Case 4 initiated its ML project after a request from an important customer, while Case 1 aimed at allowing anglers to get insights into where they should fish.

The importance of clear objectives and goals is illustrated by the fact that the Case 1 organization had experimented and tried a few similar proof-of-concepts (POC) but never managed to get something at large scale into production. When Modulai, the ML consultancy firm, was contacted, one of their first actions was to boil the hypothesis down into critical questions and develop clear objectives and goals. The ML engineer explained: *"We often help define the key questions with the client, to understand which problem we are trying to solve concretely."* In discussion with the Case 1 organization, they set a clear goal of trying to raise the rate of catches for the users and thereby increase conversion to their premium subscription service. It was only after these objectives were specified that the project moved on to develop and later successfully implement the ML system. When asked whether the process of setting objectives truly is critical to the projects, he replied: *"You must formulate a question, and this can often lead to misunderstandings. If you are not super clear, it is difficult."*

Setting clear objectives and goals seems to be a CSF for ML implementation projects as efforts to implement ML never succeeded without them.

4.1.3 Setting up requirements

Valuing time experimenting on the project rather than determining requirements is something shared by the successful cases. For instance, in Case 4, once the initial idea was introduced, a POC was quickly created by using an API to a pre-trained model and then trained the model on some annotated speech data. The organization did no other planning than assigning two full-time developers to improve on the demo system.

The internal data provided by the Case 1 organization was well structured. However, according to the project leader, it was still not certain if it was enough to work as a general recommender for anglers. However, instead of trying to spend time deciding on the necessary requirements and specifications needed for the system, the team prioritized building time. The engineer in Case 1 elaborated: *"We allocated four weeks to test and see what was possible to build."* Alongside this effort, Modulai started looking into external data sources that could enhance the model.

Case 2b and 3 both had the feasibility assessment quickly made by their research departments. The project leader in Case 3 explained: *"We had done several similar projects, but now we wanted to take it a step further."* Both project leaders regarded the business cases as straightforward, as well as the challenges ahead, which indicate their experience. As the project leader explained in Case 2b*: "The data labeling was clear and image classification is easy with deep learning."* Hence, the pre-study was done in a quick manner, mostly based on the project leader's own experience, with a focus on moving forward to execution.

For Case 2b, the project team understood from a discussion with the product owner, that a low false-positive rate was significant. If high, this metric would damage the customer experience as correct behavior would not lead to the cashback, i.e., a person uploading a correct picture does not receive a reward. After the team experimented and discussed with the product owner, they realized that a threshold value of around 1% would have to be achieved in order for it to be fit for production.

While setting up requirements for the project, valuing experimentation over planning seems to be a CSF for ML implementation projects.

4.2 Planning

In the project phase Planning, one theme revealed a CSF. This theme was Scheduling of the project.

4.2.1 Scheduling of the project

An important similarity between the successful cases was the team's ability to create realistic schedules. Based on his experience from similar projects, the project leader in Case 2b estimated that five weeks would be enough for the second try of the project. As this was a short time frame, the time was allocated into a single sprint. The project leader in Case 2b explained: *"I essentially did a long todo-list of tasks and estimated the approximate time consumption of each."* The project leader in Case 3 employed a similar approach. A project group was established alongside a timeboxed schedule of 4-5 weeks with specified activities. According to the project leader, the need for planning and coordination was lower than it might be for other companies, as the company has a flexible infrastructure and transparent processes for deploying code every other day. He elaborated: *"...if other companies without these large data and infrastructure teams tried the same thing, in the same timeframe, they would have needed much more people."*

Modulai usually plans for their case-phases to take between 6-16 weeks. In Case 1, the ML engineer in charge scheduled time for both learning the context of the problem and experimentation. He explained: *"Try to work to the point where you can start first validations of the system after 1-2 weeks."* The project leader had to synchronize their scheduling with the sprint lengths and processes at the client organization.

Case 2a, in part, failed because they did not schedule and timebox their tasks accurately enough. This allowed them to exceed their budget and had after six months still not delivered any software. According to the project leader, who initiated the second attempt, Case 2b, too much time was allocated to the modeling phase.

Setting a realistic schedule seems to be a CSF in ML implementation projects due to its adherence and non-adherence between the successful and unsuccessful cases, respectively.

4.3 Execution

In the project phase Execution, three distinct themes revealed a CSF. They are Data handling, Applied project management methodologies, and Technology or user orientation.

4.3.1 Data handling

A major part of the execution for all projects was collecting, experimenting with, and refining the dataset that the model relies on for training and validation. Repeatedly highlighted was the importance of dedicating time to get a deep understanding of the dataset in use. For instance, one of the key reasons why Case 2b succeeded, as opposed to the failed Case 2a attempt, was in the treatment of data. In the first try, the project leader assigned three part-time workers to annotate the data with no modeling experience. The team members who were then supposed to build the model had no understanding of their data distribution, and especially, the corner cases. In the second try, the project leader spent a significant portion of time in personally skimming through 20,000 pictures to understand the data fully. He explained: *"The first step was to embrace the role of being the algorithm. I sat there myself and assessed 20,000 images to get a firm understanding of the data and potential corner cases."*

In another example, Modulai expressed how impressed they were with Case 1 organization's high level of data quality, which allowed them to execute faster and better: *"They were special in that sense that they realized that the data they had was unique, relevant, and structured. They are digital natives; they understand how to log the data and the value of it. Companies we otherwise come in contact with discuss ML but do not possess that kind of data."* The organization's dataset included data from over five million catches. Yet, to deepen the understanding of the data, even more, the project leader from Modulai went out in fishing boats to get a first impression of the complexities of predicting a catch.

Like the others, both Case 3 and 4 did significant work on their data. According to the project leader in Case 4, it was especially important due to their realization that particular dialects were troublesome for their speech recognition model to interpret. When asked whether this understanding was critical to the project, the AI research engineer in Case 4 said: *"It really was. We had to do a lot of customization for the dialects."* This entailed creating and using new datasets of recorded speech with phonetic transcription. For Case 3, they had done many similar projects before. Yet, they still spent a significant portion of the allocated time of their project on data management.

That one or more people in the project dedicate time for truly understanding the data sources and the corner cases is a CSF for ML implementation projects indicated by this study.

4.3.3 Applied project management methodologies

All successful cases were applying agile project management methodologies. According to the project leader in Case 1, in a typical agile fashion, the client organization and the consultants synchronized their sprints and retrospectives. In addition, they tried to reach an end-to-end working proof of concept before honing in on and improving any specific task in the process.

This process was similar to the one followed in Case 4. The project leader explained: "*We do daily standups every day, and planning on Fridays. Then we follow kanban boards to plan a project. There is one board per project."* In addition to using digital Kanbans to keep up with tasks that needed to be performed, the team met weekly with the rest of the organization to align their progress.

In Case 3, according to the project leader, rapidly delivering value was a guiding principle. Furthermore, he specifically argued for the organization's CEO's strategy of "failing fast", meaning the result of an innovative project should reveal its value quickly.

In Case 2a, the project was not following effective project management methodologies, and certainly not an agile framework. As the work was not sufficiently timeboxed and, according to the project leader, the team never delivered any value, ineffective project management methodologies seem to play a large part in the project failure. According to the Case 2b project leader, they never managed to deliver any complete code. He explained: *"After nine months when I took over as project leader, we had to start all over again."*

For the second try with the spam recognition, Case 2b, the time was so short that the project manager did not bother with any formal agile procedures besides working in sprints. However, this was enabled by the fact that they were a very experienced team in agile software development. During the five weeks, no meetings were scheduled, and the team instead did one-on-one syncs as they stumbled upon problems with their tasks. The project leader had a clear view of the primary objective: *"Up in production as quickly as possible, then you get a grasp of how it performs. That was one of the problems with what they had not done from the earlier project."*

Effective project management methodologies, and especially agile ones, seem to be a CSF for ML implementation projects as this is followed by the successful cases, but not in Case 2a.

4.3.4 Technology or user orientation

In the execution phase of the project, a distinct focus on solving the problem for the end-user seemed to be a vital aspect of success. This focus implied at times to make the model more complex for the customer, but at times to limit the complexities of the model to a large degree.

The latter case was especially apparent in Case 1. The team noticed that the model predicted sharp spikes in the probability of a certain fish catch in certain places during the day. Even though on average, this would have been more accurate, the team chose to make it a little less accurate, to smooth down the probability curves, and make it fit the user experience and design language. The project leader in Case 1 elaborated: *"The goal was not to maximize the model's performance but to receive great feedback from customers. We wanted the customer to feel: "I have used your service and think it is great.""*.

The solution focus enabled Case 3 to deliver valuable insights to the company, even though it failed according to this thesis's success criteria. According to the project leader, the team worked hard on including potential variables that could contribute to retention, so-called feature engineering. From feature engineering around 40 different features, it turned out only one was relevant when the model was tested. The project leader explained: *"Then we aimed to see which features had the most predictive power if a user would be retained or not. Out of all features we engineered, which took quite some time to do, we found that it was only one which was heavily correlated with if an end-user would be retained."* This finding turned the ML project on its head. With merely one feature relevant, it would be more appropriate to hard code solutions to increase retention rate, rather than to deploy their model. In the end, they chose not to use their system and instead hard-coded rules based on this single feature.

Interestingly, the ML engineer from Case 1 touches upon the same idea: *"Sometimes the coolest AI project becomes a linear regression. Super basic, but it can be the best for that specific project"*. Hence, the chosen model to solve the problem is mainly dependable on how it should be used by the end customer, internal or external.

Contrary to the project group in Case 1 and 3, whom both chose a less technically complex solution for the actual deployment, Case 4 had to do the opposite. To speed up the pronunciation assessment, the team had to switch from CPU to GPU inference. The project leader elaborated: *"It took a while to* *solve the inference on the GPUs, and not just train on GPUs. But when we did, we got much faster answers which were important to the customer."*

Being able to prioritize the solution over the specific technical method is a CSF in ML implementation projects. Repeatedly, different aspects of this CSF were illustrated, both to reduce and increase complexity with the end solution in mind.

4.4 Finalization

In the project phase Finalization, monitoring of models was discussed, but the phase uncovered no CSF. The successful implementations continuously monitor the models' performance. Nevertheless, long term monitoring is considered outside of the scope of this study.

5. Analysis

In this section, the themes that reveal CSFs are analyzed by contrasting and comparing the findings to previous research in order to enable a deeper understanding of the findings.

5.1 Idea & pre-study

5.1.1 Clear objectives and goals

All the projects included in the study followed the recommendation of Pinto (2016 p.150-153) with clear goals and objectives set quickly already in the idea and pre-study phase. The importance of this factor was clearly indicated in Case 1. There, several experiments and proof of concepts had not led to a model in production, and it was only when the ML consultants helped them truly refine their ideas into clear objectives and goals that the project succeeded.

The practice of refining the critical questions observed in the cases is following Heagney's (2010 p.43) advice for project planning. This is also aligned with the CSF "Clear objectives and goals" that is outlined in previous research (Nasir & Sahibuddin, 2011; Keil et al., 2002; Schmidt et al., 2001). In Reels (1999) influential paper, he wrote that one must also diffuse these goals of the software project to all relevant stakeholders in order to manage expectations. This is precisely what happened in Case 1, as the development team refined the clear objectives and continuously aligned with the client team. He explained: *"Our methodology is always in step zero to have an intensive workshop with the people we will work with to ensure that they understand our way of thinking."*

Pinto (2016 p.153) writes that the objectives and goals should be measurable, tangible, and verifiable. This is validated in this thesis, as well. It is evident that clear goal setting is not enough; however, as objectives and goals were identical in Case 2, yet it failed the first time but not the second.

5.1.2 Experimentation over planning

All successful cases minimized time specifying detailed requirements, and instead focused on getting a fully functioning system up and running. The successful cases adhere to what Tonnquist (2016 p.93-95) would classify as projects without project-requirements. Then, they started experimenting to see what was possible. As the AI research engineer in Case 4 explained: *"..to get a week to test out a project is generally beneficial. Often, it does not work. Nevertheless, at times, it does. If you do not*

plan that much in beforehand, but instead start working on getting the easiest possible project up, you get much more information on what could work and the opposite to plan what is needed and how much time it will take." This resulted in a comparatively short pre-study, mostly based on the project leader's own experience, with a focus on moving forward to execution.
Clear and frozen requirements are a commonly recurring success factor in the software CSF literature (Kappelman et al., 2006; Schmidt et al., 2001). Without frozen requirements, the system will never go into production as it continually is in development, and therefore the recommendation is to keep the requirements stable (Schmidt et al., 2001). However, this line of thinking assumes that most of the requirements are known and that it is possible to freeze them. In the successful cases of this study, this is not what is observed. The following passage from one of the interviews in Case 1 illustrates: *"We allocated four weeks for testing and seeing what was possible to build."* Current ML focused literature is starting to reveal similar results, where Arpteg et al. (2018) argues that the performance of a system is unknown until it has been trained on the right amount of data, making planning difficult.

As this means minimized time spent on planning or determining requirements, it raises the question about how to prioritize and evaluate which projects are feasible and worthwhile pursuing. Here, successful cases can provide insights. Both the ML engineer at Modulai and the project leader from the e-commerce platform looked at results from academia and known industry use cases to quickly estimate feasibility.

Furthermore, the CSF is also exemplified in the failed first try at the e-commerce firm in Case 2a. The first time, the project group had spent the majority of their time on the pre-study to discuss modeling and how to reach the highest accuracy, as opposed to experimenting and testing with data labeling and deployment. The project leader in Case 2b explained: *"The problem with the first try of the project was that the team members focused on how to do the modeling perfectly, rather than to actually do it."*

CSF	**ML CSF**	**Software CSF**
Clear objectives and goals	Yes	Yes
Experimentation over planning	Yes	

Table 4. CSFs from Idea & pre-study phase

5.2 Planning

5.2.1 Realistic schedule

Tonnquist (2016 p.157-163) argues that projects need to specify their resources and schedule in the planning phase of the project. The successful cases all adhere to this recommendation and manages to estimate resource and time consumption accurately, despite the difficulties of ML modeling. Furthermore, this CSF is frequently observed in the software development literature (Humphrey, 2005; Taylor, 2006; Reel, 1999; Nasir & Sahibuddin, 2011; Kappelman et al., 2006).

Case 2a, in part, failed because they did not schedule and timebox their tasks accurately enough, and they ended up after six months without any delivered code. According to the project leader, who initiated the second attempt, too much time was allocated to the modeling phase of the project.

According to Kappelman et al. (2006), a common way of scheduling in projects is to list individual tasks and estimate their time. The project leader in Case 2b did just that: *"I essentially did a long todo-list of tasks and estimated the approximate time consumption of each."* This was repeated in the interview for Case 3, who also specified activities in a four-week schedule.

As the failed Case 2a did not fulfill these requirements, it would suggest that a "Realistic schedule" is equally essential in software implementation and ML implementation projects alike. Naturally, this raises the question of how to accurately construct such a schedule. Jones (2006) and Pinto (2016 p.322) argue that project estimation and planning often can be helped by different software programs. This is not observed in the successful cases. Instead, all projects are estimated based on the project team's experience from similar projects.

CSF	**ML CSF**	**Software CSF**
Realistic schedule	Yes	Yes

Table 5. CSFs from planning phase

5.3 Execution

5.3.1 Deep understanding of the dataset

Project managers need to grasp the context, prerequisites, and dependencies in order to deliver a successful project. Understanding the dependencies is important throughout the phases of the project, but especially important to manage during the execution phase (Tonnquist, 2016 p.307-311). External data introduces a dependency on ML projects, which differ from most other software projects, as the outcome is fundamentally determined by the quality and quantity of the data. This is because the data is used to program the system rather than writing the code manually (Arpteg et al., 2018). Therefore, one could see data as a resource in line with Tonnquists (2016 p.307-309) argumentation.

Arpteg et al. (2018) give several examples of ML specific dependencies. For example, the sheer volume of the data can require a distributed solution to storage and computation. This adds complexity and requires additional knowledge into managing these systems (Arpteg et al., 2018).

These new dependencies presented by Arpteg et al. (2018) introduces new tasks in the execution phase of projects, which implies the need for additional competencies in the project team as well. For instance, in their framework for ML implementations, Zhou et al. (2017) argue that the "Data Preprocessing" stage includes seven distinct challenges. Managing this stage of the ML project effectively clearly contributes to the success of the projects. This is also what is observed in the successful cases in the study. All successful cases arranged significant time towards understanding their data, which is not required in regular software implementation projects. The project leader in Case 2b personally made sure 20 000 pictures were correctly labeled. In Case 1, the project leader went out into a fishing boat to understand the process of data creation and what other data sources might be relevant to his problem. The project leader in Case 1 elaborated: *"..going from raw data to finished features, that is the lion's share of the work."* That data handling is time-consuming is also a fact reflected in the ML implementation literature by authors such as Bengio et al. (2013).

These strenuous, yet significant, tasks are clearly exemplified in the cases involved in the study. However, the project leaders in the successful cases worked proactively with the most up to date tools, and all companies had their infrastructure completely cloud-based, which facilitates computation and storage.

5.3.2 Effective project management methodologies

Pinto (2016 p.370) argues that the underlying assumption of traditional project management methods is twofold: minimal uncertainty and maximum stability. In a more unpredictable environment, Agile is better suited, and it is therefore often applied in IS-development projects (Pinto, 2016 p.370-371; Tonnquist, 2016 p.99). All case organizations included in the study were applying Agile project management in their workstreams. The successful cases all reap the benefits from this methodology of early delivery, ability to handle changes, and action despite that the requirements are unclear. The process for early delivery was clearly illustrated by the project leader in Case 1, who said: *"Try to arrive at a state where you can validate [the model] within 1-2 weeks. Then you can iterate it."*

Some researchers, such as Chow & Cao (2008) and Highsmith (2002), imply that Agile project management has its own distinct set of CSFs. However, Tonnquist (2016 p.39) argues that fundamentally, Agile methodologies and regular project management methods share many common characteristics. Instead, he argues that Agile methodologies differ mostly in terms of the processes in the execution phase of the project.

This CSF is frequently recurring in the software development literature as well (Schmidt et al., 2001; Keil et al., 2002). Our findings, therefore, indicate that ML projects and regular software implementation projects are both in need of effective project management.

5.3.3 Solution over technology orientation

The successful cases all had a clear and unabated focus on the solution most suitable for the end-user, rather than focusing excessively on the technology itself. This resulted in that two projects simplified their model to suit their end-user, while one had to make it more complex to reach a satisfactory user experience.

This solution focus goes in line with what Tonnquist (2016 p.99) would classify as early product development projects. In these projects, the technical requirements and details are not set but are exchangeable in order to meet the end-customers preferences. In addition, the specifications are developed into more detail as the team gains knowledge and experience from trying and testing the product (Tonnquist, 2016 p.99-100).

Previous CSF studies in a software development context have not found this CSF to be critical in their projects. Yet, it repeatedly showed its importance among the cases. In Case 1, they decided to reduce the accuracy of the model to fit with the user experience. In Case 3, the project leader decided to abandon the project entirely and instead code a few simple rules. Both cases made the solution less technologically advanced, in favor of the end-user. The opposite happened in Case 4, where the project team made the solution more advanced to meet the customers' need.

Many other studies have found that end-user/client involvement is a CSF for software projects (Sauer & Cuthberson, 2003; Standing et al., 2006; Charette, 2005). The cases in this study do not necessarily point to the active involvement of the end-user, but rather that their user experience is prioritized. This is more aligned with the CSF of meeting customer expectations and the Agile principle of listening to customers (Shenhar et al., 1996; Pinto, 2016 p.372). Despite the aforementioned similarities, it raises the question of why solution over technology orientation seems more important in ML implementation projects than what previous studies have found.

A potential reason for this discrepancy might be attributable to a difference between academia and building customer-ready products. In Case 1 and Case 2, the project leaders reflected on a common issue. From Case 1: *"Many companies just hire a recent engineering physics graduate. But there is a large difference between making models in school versus the real world. Platforms, dirty data, and collaboration, that is not something you do in school."* When asked why being solutions-oriented seems so important for ML projects, the project leader in Case 2 explained that in his experience, many ML engineers tend to focus too much on technology. He elaborated: *"An overarching trend in ML is that people working on it come directly from academia and want to build cool models, as opposed to products. There is a limited business mindset. In academia, it is about an interesting model that should be developed. [..], but these models are not always reasonable for products."*

A hypothesis for why this CSF is prevalent in this study, as opposed to earlier software implementation studies, is that ML has relatively recently found broad practical implications. Therefore, practitioners might not yet have the same user focus as software engineers in general.

CSF	**CSF in ML**	**CSF in Software**
Deep understanding of dataset	Yes	
Effective project management methodologies	Yes	Yes

Solution oriented over technology oriented	Yes	

Table 6. CSFs from Execution phase

5.4 Finalization

No CSF emerged from the finalization phase of the projects. Our research focuses on how to plan and execute the project to deploy the model successfully. According to Sculley et al. (2015), a deployed model always requires maintenance, which was briefly touched upon in several cases. However, monitoring the ML model for the long term is out of the scope of this thesis.

CSF	**CSF in ML**	**CSF in Software**
Clear objectives and goals	Yes	Yes
Experimentation over planning	Yes	
Realistic schedule	Yes	Yes
Deep understanding of the dataset	Yes	
Effective project management methodologies	Yes	Yes
Solution over technology orientation	Yes	

Table 7. All CSFs found in the cases

6. Discussion

The findings analyzed in the previous section are discussed more deeply to reveal both the similarities and differences between ML and regular software implementation. In addition, the contribution to both practitioners and academia are discussed.

6.1 Contribution of the study

The result of this study is a list of CSFs essential for successful ML implementations in software organizations. The proposed list contributes to closing the research gap around implementing ML in organizations on the group level of analysis, as previous CSF research has been done for AI on an organizational level. The analysis illustrated certain similarities and differences between current research in software implementations, and the findings of the study. These are elaborated upon and discussed below.

6.1.1 The similarities between ML and software implementations

The CSFs “Clear objectives and goals”, “Effective project management methodologies”, as well as “Realistic schedule”, were found to be highly relevant for both ML implementation projects and software implementation projects in line with Nasir & Sahibuddins (2011) review. A probable cause of this similarity is that they form what Tonnquist (2016 p.16-22) argues, part of the foundations of project management, regardless of the specific project domain.

Arguably, “Experimentation over planning”, “Solution over technology oriented” and “Effective project management methodologies” all connect in different ways to Agile project management. The authors motivate the reason for describing these as separate CSFs due to the fact that the benefits of Agile methodologies are still under discussion, as argued by Pinto (2016 p.376). Serrador & Pinto (2015) showed that the degree of Agile implementation applied in a project is impacting its project success. Therefore, by showing the most important aspects relating to Agile development as specific factors, the critical aspects of this methodology is brought forward and discussed in-depth.

6.1.2 Where software and ML implementation differ

In addition to the CSFs that are similar, the results of the study indicate three new CSFs specific for ML implementation projects. They were "Deep understanding of the dataset", "Experimentation over planning", and "Solution over technology-oriented".

Deep understanding of the dataset highlights that ML introduces an element of external data sources to the software project. As the system no longer consists of hard-coded rules, the success of the project is largely dependent on the quantity and quality of the data.

This study's findings indicate that solutions-oriented projects, rather than technology-oriented, succeeded. This difference was highlighted by the successful cases where project groups either expanded or contracted technical complexity depending on the feedback from the end-user. Both Case 1 and Case 2 raised awareness of this CSF as contributing when project groups wasted time caught up dabbling with technical questions. Contradictory incentives between ML practitioners wanting to publish their technical accomplishments with product owners wanting to deliver value was considered a potential contributing factor.

Lastly, whereas software implementation projects require a thorough pre-study, with most details clarified in the beginning, this study reveals that experimentation is valued over planning. The difficulty of estimating specific parameters of the final output made it more fruitful to try, experiment, and validate the system. Therefore, "Experimentation over planning" is a CSF that clearly differs from previous software implementation research. This interesting contradiction is further expanded below.

6.1.3 Where software and ML implementation contradict

Nasir & Sahibuddins (2011) review revealed that "Clear requirements and specifications" are considered one of the most important critical success factors of software implementation. However, in all cases, the difficulty of estimating an evaluation metric and, therefore, the performance of the system was brought up. This was exemplified by the ML engineer from Case 1, who said: *"We can estimate the time and budget, but no amount of compute or data can guarantee an outcome without testing it. We are very transparent about that."*

This result stands in stark contrast in comparison with previous literature on projects in a software context. It suggests that “Clear requirements and specifications” is not a CSF for these types of projects. Naturally, this raises the question of why.

All successful projects quickly moved on from their feasibility study towards execution. These projects prioritized speed and experimentation rather than spending time specifying requirements in line with Agile methodologies. However, these practices seem even more important in ML implementation than in Agile software development. This is because the performance of the system depends on the plethora of choices the ML engineer makes in setting up the model, the dataset, and even certain stochasticity in training itself, in line with what Arpteg (2018) and Goodfellow et al. (2016 p.54) have argued. In the failed project for the media streaming service, they even abandoned the ML system altogether, as a much simpler indicator, based only on one parameter, predicted the outcome equally well. Nevertheless, that result would never have been uncovered if they had not developed the ML model from the beginning.

6.2 Contribution to practitioners

Looking at the similarities and differences of CSF in this study will empower practitioners of project management within the context of ML implementations by contributing with highly applicable knowledge. It presents managers with a more focused set of success factors specifically tailored for ML, which is important as many managers argue, according to Ransbotham et al. (2017) that AI, is becoming a competitive advantage for their organizations. By studying the similarities and differences between the ML cases from this study, and discussing them with the team, organizations might be less prone to get stuck in ideation phases of their ML initiatives.

7. Conclusion

The purpose and goal of this study have been to further the understanding of project management and its connections to AI, specifically ML, by answering the research question: *"What are the critical success factors for implementing machine learning for software companies?"* Based on a multiple case study of four organizations in the software industry, including both failed and successful implementations, the study reveals six CSFs for ML implementation projects. "Clear objectives and goals", "Effective project management methodologies" and "Realistic schedule" were CSFs in both regular software implementation projects and in ML implementation. "Deep understanding of the dataset", "Experimentation over planning" and "Solution over technology oriented" were all new ML specific CSFs that project managers should take into account in their next projects.

The study complements the previous research on a group level of analysis within management by extending current CSF research in software implementation to ML specific CSFs.

7.1 Generalizability & limitations of the study

The interviewees, during the interview occasions, could have intentionally or unintentionally omitted valuable information relevant to the outcome of the project. Reasons for this include them being biased towards the importance of their own performance or aspiring not to talk bad about the organization, as has been argued by Bell et al. (2019 p.458-459). However, by using multiple cases, this study tries to ensure a representative selection and robust findings.

Even though this study can significantly add to the project understanding of ML implementations, it should be noted that other cases might have yielded some different success factors. This study does not propose to have delivered an exhaustive list of all CSFs, but rather the ones that our selection of cases presented.

On a final note, this study's generalizability to companies outside the software industry might be hampered by their IT infrastructure. To validate this proposition, the authors of the study also interviewed an engineer involved in a non-software company, which illuminated several barriers, such as building a dataset and a pipeline for acquiring more data as well as deploying the model.

7.2 Proposal for further research

Future research could explore the validity and generalizability of the proposed CSFs discovered in this study for software organizations. The list of CSFs generated in this study could be tested by project managers in new projects and studied to ensure their validity. To further strengthen the findings, they should be validated through a larger quantitative survey from a probability sample, as this would determine the generalizability of the findings.

Furthermore, research should also be done on organizations in other industries to deepen the understanding of challenges and solutions for project managers acting in these fields. One reason for this, as mentioned in the empirics, is that the quality of IT infrastructure varies greatly in between industries.

8. References

Ahmed, Z., Kumar, U. & Kumar, V. 2018, "Managing critical success factors for IS implementation: A stakeholder engagement and control perspective", *Canadian Journal of Administrative Sciences / Revue Canadienne des Sciences de l'Administration,* vol. 35, no. 3, pp. 403-418.

Alhassan, I., Sammon, D. & Daly, M. 2019, "Critical Success Factors for Data Governance: A Theory Building Approach", *Information Systems Management,* vol. 36, no. 2, pp. 98-110.

Amazon Web Services 2020, *Modulai used Amazon SageMaker to Build the New Models for Fishbrain's Fishing Forecast*. Available: https://aws.amazon.com/partners/success/fishbrain/ [2020, 19 Feb].

Arpteg, A., Brinne, B., Crnkovic-Friis, L. & Bosch, J. 2018, "Software Engineering Challenges of Deep Learning", *44th Euromicro Conference on Software Engineering and Advanced Applications*, Prague, Aug 2018, pp. 50-59.

Baccarini, D. & Collins, A. 2003, "Critical success factors for projects", *Proceedings of the 17th ANZAM Conference*, Fremantle, Dec 2003.

Belassi, W. & Tukel, O.I. 1996, "A new framework for determining critical success/failure factors in projects", *International Journal of Project Management,* vol. 14, no. 3, pp. 141-151.

Bengio, Y., Courville, A. & Vincent, P. 2013, "Representation learning: A review and new perspectives", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 8, pp. 1798-1828.

Boehm, B. 1991, "Software Risk Management: Principles and Practices", *IEEE Software,* vol. 8, no. 1, pp. 32-41.

Boynton, A. & Zmud, R. 1984, "An Assessment of Critical Success Factors", *Sloan Management Review (pre-1986),* vol. 25, no. 4, pp. 17.

Bredillet, C.N. 2008, "Exploring Research in Project Management: Nine Schools of Project Management Research (Part 4)", *Project Management Journal,* vol. 39, no. 1, pp. 2-6.

Brown, S.L. & Eisenhardt, K.M. 1997, "The Art of Continuous Change: Linking Complexity Theory and Time-paced Evolution in Relentlessly Shifting Organizations", *Administrative Science Quarterly,* vol. 42, no. 1, pp. 1.

Bryman, A., Bell, E., Harley, B. 2019, *Business research methods,* 5th edn, Oxford Univ. Press, Oxford.

Burrel, G. & Morgan, G. 1979, *Sociological Paradigms and Organisational Analysis,* 1st edn, Ashgale Publishing Company, Burlington.

Cambridge Dictionary. 2019,*Cambridge Dictionary Algorithm* 2019a, b-last update. Available: https://dictionary.cambridge.org/dictionary/english/algorithm [2020, February 25,].

Cambridge Dictionary. 2019, *Cambridge Dictionary System* 2019b, a-last update. Available: https://dictionary.cambridge.org/dictionary/english/system [2020, February 25,].

Carlton, D. 2017, "Competence VS Confidence in IT Project Leadership and its Impact on Project Outcomes", *Journal of Modern Product Management,* vol. 5, no. 1, pp. 38-45.

Cassell, C., Cunliffe, A. & Grandy, G. 2017, *The SAGE Handbook of Qualitative Business and Management Research Methods,* 1st edn*,* SAGE, London.

Charette, R.N. 2005, "Why software fails", *IEEE Spectrum,* vol. 42, no. 9, pp. 36-43.

Chow, T. & Cao, D. 2008, "A survey study of critical success factors in agile software projects"*,The Journal of Systems and Software,* vol. 81, no. 6, pp. 961-971.

Cooke-Davies, T. 2002, "The "real" success factors on projects", *International Journal of Project Management,* vol. 20, no. 3, pp. 185-190.

Croteau, A. & Li, P. 2003, "Critical Success Factors of CRM Technological Initiatives", *Canadian Journal of Administrative Sciences / Revue Canadienne des Sciences de l'Administration,* vol. 20, no. 1, pp. 21-34.

Cuthbertson, C. & Sauer, C. 2003, "The State of IT Project Management in the UK 2002-2003", *Computer Weekly,* 15 April, 2003.

Deakin, H. & Wakefield, K. 2014, "Skype interviewing: reflections of two PhD researchers", *Qualitative Research,* vol. 14, no. 5, pp. 603-616.

Devlin, J., Chang, M., Lee, K. & Toutanova, K. 2019, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of 17th North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, June 2019, pp. 4171-4186.

Diener, E. & Crandall, R. 1978, *Ethics in social and behavioral research,* 1st edn, Univ. of Chicago Press, Chicago.

Dyer, W.G. & Wilkins, A.L. 1991, "Better Stories, Not Better Constructs, to Generate Better Theory: A Rejoinder to Eisenhardt", *Academy of Management Review,* vol. 16, no. 3, pp. 613-619.

Fairley, R.E. 2009, *Managing and leading software projects,* 1st edn, John Wiley & Sons, New Jersey.

Flyvbjerg, B. 2006, "Five misunderstandings about case-study research", *Qualitative inquiry,* vol. 12, no. 2, pp. 219-245.

Forbes. 2018, "*Okay Google: Voice Search Technology And The Rise Of Voice Commerce*", Available: https://www.forbes.com/sites/tjmccue/2018/08/28/okay-google-voice-search-technology-and-the-rise-of-voice-commerce/#c51cbbd4e292 [2020, May 9].

Fortune, J. & White, D. 2006, "Framing of project critical success factors by a systems model", *International Journal of Project Management,* vol. 24, no. 1, pp. 53-65.

Fountaine, T., McCarthy, B. & Saleh, T. 2019, "Building the AI-Powered Organization". *Harvard Business Review.* [online]. Available: https://hbr.org/2019/07/building-the-ai-powered-organization May 9].

Frese, R. & Sauter, V. 2003, "Project success and failure: What is success, what is failure, and how can you improve your odds for success", *St. Louis, Missouri.*

Gandhi, P., Khanna, S. & Ramaswamy, S. 2016, "Which Industries Are the Most Digital (and Why)?", *Harvard Business Review, [Online],* Available: https://hbr.org/2016/04/a-chart-that-shows-which-industries-are-the-most-digital-and-why

Ghosh, S. & Skibniewski, M.J. 2010, "Enterprise resource planning systems implementation as a complex project: A conceptual framework", *Journal of Business Economics and Management,* vol. 11, no. 4, pp. 533-549.

Giertz, E., Levén, P., Sörensson, R. & Lougui, M. 2013, *Företag inom informations och kommunikationsteknik i Sverige 2007 - 2011*. Available: https://www.vinnova.se/contentassets/187e13404eaf4b2c8d4646f331fe2994/va_13_07.pdf

Goodfellow, I., Bengio, Y. & Courville, A. 2016, *Deep Learning,* 1st edn, MIT press, Cambridge, MA.

Gustavsson, T. 2019, *Agile Project Management,* 1st edn, Sanoma utbildning, Stockholm.

Haenlein, M. & Kaplan, A. 2019, "A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence", *California Management Review,* vol. 61, no. 4, pp. 5-14.

Hanna, P. 2012, "Using internet technologies (such as Skype) as a research medium: a research note", *Qualitative Research,* vol. 12, no. 2, pp. 239-242.

Heagny, J. 2011, *Fundamentals of Project Management,* 4th edn, Amacon, Saranac Lake.

Henderson, J.C., Rockart, J.F. & Sifonis, J.G. 1987, "Integrating Management Support Systems into Strategic Information Systems Planning", *Journal of Management Information Systems,* vol. 4, no. 1, pp. 5-24.

Herriott, R.E. & Firestone, W.A. 1983, "Multisite Qualitative Policy Research: Optimizing Description and Generalizability", *Educational Researcher,* vol. 12, no. 2, pp. 14-19.

Highsmith, J.A. 2002, *Agile software development ecosystems,* 1st edn, Pearson Education, Boston, MA.

Hosanagar, K. & Saxena, A. 2017, "The First Wave of Corporate AI Is Doomed to Fail". *Harvard Business Review. [Online]* Available: https://hbr.org/2017/04/the-first-wave-of-corporate-ai-is-doomed-to-fail [2020, May 9].

Humphrey, W.S. 2005, *Why big software projects fail: The 12 key questions,* 7th edn, John Wiley & Sons Inc, Hoboken, New Jersey.

Inside Big Data. 2018, *Amazon Go – Deep Learning Conquers Retail 2018a*. Available: https://insidebigdata.com/2018/02/15/amazon-go-deep-learning-conquers-retail/ [2020, May 9].

Jain, M. 2008, *Delivering Successful Projects with TSP (SM) and Six Sigma: A Practical Guide to Implementing Team Software Process (SM),* 1st edn, Auerbach Publications, Boca Raton.

Jones, C. 2006, "Social and technical reasons for software project failures", *CrossTalk,* vol. 19, no. 6, pp. 4-9.

Jones, N. 2014, "The Learning Machines", *Nature,* vol. 505, no. 7482, pp. 146.

Kahn, J., *Inside big tech's quest for human-level A.I.* Available: https://fortune.com/longform/ai-artificial-intelligence-big-tech-microsoft-alphabet-openai/ [2020, May 9].

Kai-Uwe Brock, J. & von Wangenheim, F. 2019, "Demystifying AI: What Digital Transformation Leaders Can Teach You about Realistic Artificial Intelligence". *California Management Review,* vol. 61, no. 4, pp. 110-134

Kamal, M.M. 2006, "IT innovation adoption in the government sector: identifying the critical success factors", *Journal of Enterprise Information Management,* vol. 19, no. 2, pp. 192-222.

Kappelman, L.A., McKeeman, R. & Zhang, L. 2006, "Early warning signs of IT project failure: The dominant dozen", *Information Systems Management,* vol. 23, no. 4, pp. 31-36.

Karlsen, J.T., Andersent, J., Ødegård, E. & Birkely, L.S. 2005, "What characterizes successful IT projects", *International Journal of Information Technology and Decision Making,* vol. 4, no. 4, pp. 525-540.

Kathleen M. Eisenhardt 1989, "Building Theories from Case Study Research", *The Academy of Management Review,* vol. 14, no. 4, pp. 532-550.

Keil, M., Tiwana, A. & Bush, A. 2002, "Reconciling user and project manager perceptions of IT project risk: a Delphi study 1", *Information Systems Journal,* vol. 12, no. 2, pp. 103-119.

Khanna, R. & Awad, M. 2015, *Efficient Learning Machines,* 1st edn, Apress, Berkeley, CA.

Kieser, A. & Nicolai, A.T. 2005, "Success Factor Research: Overcoming the Trade-Off Between Rigor and Relevance?", *Journal of Management Inquiry,* vol. 14, no. 3, pp. 275-279.

King, N., Horrock, C. & Brooks, J. 2019, *Interviews in Qualitative Research,* 2nd edn, Sage Publications, London.

Koc, T. 2007, "Organizational determinants of innovation capacity in software companies", *Computers & Industrial Engineering,* vol. 53, no. 3, pp. 373-385.

Langdridge, D. & Hagger-Johnson, G. 2009, *Introduction to Research Methods and Data analysis in Psychology,* 2nd edn, Pearson Education, Harlow.

Larsen, M.A. & Myers, M.D. 1999, "When success turns into failure: a package-driven business process re-engineering project in the financial services industry", *Journal of Strategic Information Systems,* vol. 8, no. 4, pp. 395-417.

Lexico, 2020a. *Lexico definition API* 2020a, a-last update. Available: https://www.lexico.com/definition/api [2020, February 19].

Lexico, 2020b. *Lexico definition CPU* 2020b, b-last update. Available: https://www.lexico.com/definition/central_processing_unit [2020, February 19].

Lexico, 2020c. *Lexico definition Project* 2020c, c-last update, Available: https://www.lexico.com/en/definition/project [2020, February 19].

Lee, B., Collier, P.M. & Cullen, J. 2007, "Reflections on the use of case studies in the accounting, management and organizational disciplines", *Qualitative Research in Organizations and Management: An International Journal,* vol. 2, no. 3, pp. 169-178.

Lim, C.S. & Mohamed, M.Z. 1999, "Criteria of project success: an exploratory re-examination", *International Journal of Project Management,* vol. 17, no. 4, pp. 243-248.

Lofland, J., Snow, D.A., Anderson, L. & Lofland, L.H. 1995, *Analyzing Social Setting: A Guide to Qualitative Observation and Analysis,* 4th edn, Wadsworth Publishing, Belmont.

Marsland, S. 2014, *Machine learning: an algorithmic perspective,* 2nd edn, CRC press, Boca Raton.

Mats Alvesson & Dan Kärreman 2007, "Constructing Mystery: Empirical Matters in Theory Development", *The Academy of Management Review,* vol. 32, no. 4, pp. 1265-1281.

Mcafee, A. & Brynjolfsson, E. 2012, "Big Data: The Management Revolution", *Harvard business review,* vol. 90, no. 10, pp. 61-68.

Mills, A.J., Durepos, G. & Wiebe, E. 2010, *Encyclopedia of case study research,* 1st edn,, SAGE Publications, Los Angeles.

Modulai. 2020, *Forecasting model predicting the success of fishing trips for millions of Fishbrain users*. Available: https://modulai.io/cases/global-forecasting-model-for-worlds-largest-fishing-app/ [2020, Feb 19,].

Moraveck, C. 2013, *Unmasking Project Management The Business Perspective of Information Systems Success,* 1st edn, Palgrave Macmillan US, New York.

Müller, R. & Jugdev, K. 2012, "Critical success factors in projects", *International Journal of Managing Projects in Business,* vol. 5, no. 4, pp. 757-775.

Munappy, A., Bosch, J., Olsson, H.H., Arpteg, A. & Brinne, B. 2019, "Data Management Challenges for Deep Learning", *45th Euromicro Conference on Software Engineering and Advanced Applications*, Kallithea, Aug 2019, pp. 140-147.

Nah, F.F., Zuckweiler, K.M. & Lee-Shang Lau, J. 2003, "ERP Implementation: Chief Information Officers' Perceptions of Critical Success Factors", *International Journal of Human-Computer Interaction,* vol. 16, no. 1, pp. 5-22.

Nandhakumar, J. 1996, "Design for success?: critical success factors in executive information systems development", *European Journal of Information Systems,* vol. 5, no. 1, pp. 62-72.

Nasir, N., Sahibuddin. "Critical success factors for software projects: A comparative study", *Scientific research and essays,* vol. 6, no.10, pp. 2174-2186

Palys, T. 2008, "Purposive sampling", *The Sage encyclopedia of qualitative research methods,* vol. 2, no. 1, pp. 697-698.

Pinto, J. 2016, *Project Management: Achieving Competitive Advantage,* 4th edn, Pearson Education, New York.

Pinto, J.K. & Slevin, D.P. 1989, "Critical Success Factors in R&D Projects", *Research-Technology Management,* vol. 32, no. 1, pp. 31-35.

Pinto, J.K. & Slevin, D.P. 1988, "Project success: definitions and measurement techniques", *Project Management Journal*, vol. 19, no. 1, pp. 67-72.

PMI 2017, *Pulse of the Profession*. Available: https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2017 [2020, May 9]

Polyzotis, N., Roy, S., Whang, S. & Zinkevich, M. May 9, 2017, "Data Management Challenges in Production Machine Learning", *ACM International Conference on Management of Data*, pp. 1723-1726.

Poon, P. & Wagner, C. 2001, "Critical success factors revisited: success and failure cases of information systems for senior executives", *Decision Support Systems,* vol. 30, no. 4, pp. 393-418.

Prem, E. 2019, "Artificial Intelligence for Innovation in Austria", *Technology Innovation Management Review,* vol. 9, no. 12, pp. 5-15.

Puaschunder, J.M. 2019, "Artificial Intelligence Market Disruption", *Proceedings of the 13th International RAIS Conference on Social Sciences and Humanities.* June, pp.1-8

Rahimi, I. & Berman, U. 2009, "Building a CSF framework for CRM implementation", *Journal of Database Marketing & Customer Strategy Management,* vol. 16, no. 4, pp. 253-265.

Raj, M. & Seamans, R. 2019, "Primer on artificial intelligence and robotics", *Journal of Organization Design,* vol. 8, no. 1, pp. 1-14.

Ramaprasad, A. & Williams, J. 1998, "The utilization of critical success factors: a profile", *Proceedings of the 29th Annual Meeting of the Decision Sciences Institute*.

Ransbotham, S., Gerbert, P., Kiron, D. & Reeves, M. 2017, "Reshaping Business With Artificial Intelligence: Closing the Gap Between Ambition and Action", *MIT Management Review,* vol. 59, no. 1, pp. n.a.

Reel, J.S. 1999, "Critical success factors in software projects", *IEEE Software,* vol. 16, no. 3, pp. 18-23.

Remus, U. & Wiener, M. 2010, "A multi‐method, holistic strategy for researching critical success factors in IT projects", *Information Systems Journal,* vol. 20, no. 1, pp. 25-52.

Rockart, J. 1979, "Chief Executives Define Their Own Data Needs", *Harvard business review,* vol. 57, no. 2, pp. 81.

Russel, S. & Norvig, P. 2010, *Artificial Intelligence, A Modern Approach,* 3rd edn, Pearson Education, New Jersey.

Ryan, G.W. & Bernard, H.R. 2003, "Techniques to Identify Themes", *Field Methods,* vol. 15, no. 1, pp. 85-109.

Rynes, S. & Gephart Jr, R.P. 2004, "From the Editors: Qualitative Research and the Academy of Management Journal", *The Academy of Management Journal,* vol. 47, no. 4 , pp. 454-462.

Schmidt, R., Lyytinen, K., Keil, M. & Cule, P. 2001, "Identifying Software Project Risks: An International Delphi Study", *Journal of Management Information Systems,* vol. 17, no. 4, pp. 5-36.

Scott, J. 1990, *A Matter of Record,* 1st edn, Polity Press, Cambridge.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J. & Dennison, D. 2015, "Hidden technical debt in machine learning systems", *In Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 2503-2511.

Serrador, P. & Pinto, J.K. 2015, "Does Agile work? — A quantitative analysis of agile project success", *International Journal of Project Management,* vol. 33, no. 5, pp. 1040-1051.

Shenhar, A.J., Renier, J.J. & Wideman, M. 1996, *"Improving PM: Linking Success Criteria to Project Type"*, 1st edn, Project Management Institute, Alberta.

Söderlund, J. 2011, *Theoretical Foundations of Project Management: Suggestions for a Pluralistic Understanding,* Oxford University Press, Oxford.

Sofian, A. 2003, "Project success in relation with organizational roles and capabilities and project managers' skills and capabilities", *PMI.*

Somers, T.M. & Nelson, K. 2001, "The impact of critical success factors across the stages of enterprise resource planning implementations", *Proceedings of the 34th Annual Hawaii International Conference on System Sciences IEEE*, vol. 8, pp. 8016.

Standing, C., Guilfoyle, A., Lin, C. & Love, P.E.D. 2006, "The attribution of success and failure in IT projects", *Industrial Management & Data Systems,* vol. 106, no. 8, pp. 1148-1165.

Su, N. 2018, "Positivist Qualitative Methods" in *The SAGE Handbook of Qualitative Business and Management Research Methods: History and Traditions* SAGE Publications Ltd, London

Su, N. 2013, "Internationalization Strategies of Chinese IT Service Suppliers", *MIS Quarterly,* vol. 37, no. 1, pp. 175-200.

Taylor, H. 2006, "Critical risks in outsourced IT projects", *Association for Computing Machinery Communications of the ACM,* vol. 49, no. 11, pp. 74.

Thirgood, J. & Johal, S. 2017, "Digital Disruption", *Economic Development Journal,* vol. 16, no. 2, pp. 25-32.

Tonnquist, B. 2016, *Projektledning,* 6th edn, Sanoma utbildning, Stockholm.

Whittaker, B. 1999, "What went wrong? Unsuccessful information technology projects", *Information Management & Computer Security,* vol. 7, no. 1, pp. 23-30.

Financial Times. 2020, *Why Tesla is taking a different approach to self-driving cars*. Available: https://www.ft.com/content/b423fe54-6cfa-11e9-80c7-60ee53e6681d [2020, May 9].

Wiener, M. 2006, *Critical Success Factors of Offshore Software Development Projects,* 1st edn, Deutscher Universitätsverlag, Wiesbaden.

Yap, C.S., Soh, C. & Raman, K.S. 1992, "Information systems success factors in small business", *Omega,* vol. 20, no. 5-6, pp. 597-609.

Yeo, K.T. 2002, "Critical failure factors in information system projects", *International Journal of Project Management,* vol. 20, no. 3, pp. 241-246.

Yeoh, W., Koronios, A. 2010, "Critical Success Factors for Business Intelligence Systems", *Journal of Computer Information Systems*, vol. 50, no. 3, pp. 23-32

Yin, R.K. 2009, *Case study research: design and methods,* 4th edn, SAGE, London

Yin, R.K. 2014, *Case study research: design and methods,* 5th edn, SAGE, London.

Zhou, L., Pan, S., Wang, J. & Vasilakos, A.V. 2017, "Machine learning on big data: Opportunities and challenges"*, Neurocomputing,* vol. 237, pp. 350-361.

9. Appendix

9.1 Introducing the term artificial intelligence and its subdomains

Today, AI is a generalizable umbrella term defined as the technique of building computers that learn and improve automatically through experience (Goodfellow et al., 2016 p.1-3). ML is a subdomain of AI that since the 1980s have gathered traction. ML models became widely adopted when working with numerical data types. However, they struggled with extracting knowledge from more complex data types, limiting managers' hope in Turing's theory of having computers performing human tasks. However, Deep learning is a modern ML technique advancing the field even further. It is loosely inspired by biological intelligence in the form of neural networks (Goodfellow et al., 2016 p.13). In the last decade, this field of ML saw exponential advancements in the performance of the models. DL has allowed computers to solve more complex problems. For instance, in 2012, an ML model shook up the entire computer science academia with extraordinary performance in computer vision (Jones, 2014). Later on, in 2018, the technology again made considerable advances in natural language processing (Devlin et al., 2019).

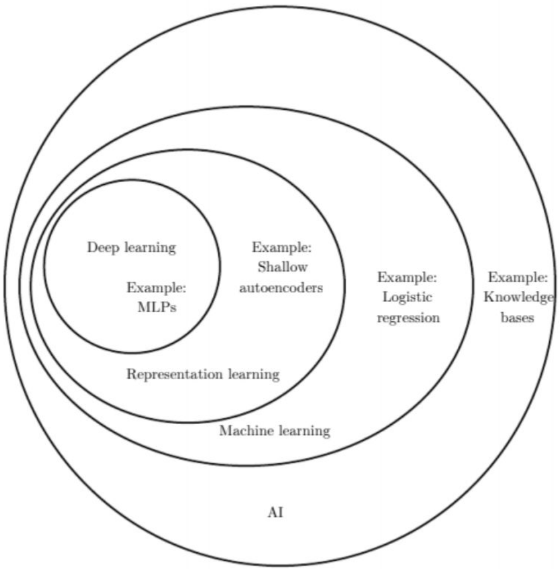

Figure 1. The field of AI and its subdomain (Goodfellow et al., 2016 p.9)

9.2 Interview guide

Ethical aspects

- If you wish, both you and the whole case can be anonymized. You can decide this up until we publish the bachelor thesis 11th of may 2020.
- Participation in this study is voluntary and you may quit the interview at any moment if you wish.
- In order to raise the accuracy and make transcribing the interview easier, is it alright if we record the interview?

Background

- What is your role in your organization?
- What was your role in the project?
- Describe the organization's general data and AI strategy

Project Idea

- What was the idea of the project?
- How essential was it for the business to solve this problem?
- Who initiated the project inside (or outside) the organization?
- What was the intended outcome of the project?

Pre-study

- How was the initial idea evaluated and turned into a business case?
- How was the feasibility of the business case assessed?
    - Did you have an intended accuracy goal of the model?

Planning

- Describe how time for the project was scheduled.
- Did you have a budget for the project?
- Describe the data acquisition process.
- Describe your project team, and how it was chosen.
- What technology suppliers, if any, did you work with?
- How was the solution considered in relation to the current software stack?
- How involved was the leadership/CEO team?

Implementation

Methodology

- Did you work with a certain method for project management?
- Did you make use of any project management tools?
- Did you follow any certain procedures when organizing the work?

Stakeholders

- How did you manage the project sponsor during the implementation?
- Could you describe the relation to the product owner regarding getting feedback?

Production

- How was the model monitored?
- Did you experience any challenges in putting the model in production?

Project Finalization

- Did the project meet the intended outcome?
- What did you bring with you to your next deep learning project?

Other

- Is there something you would like to add before we finish the interview?
- Could you recommend any other person we could talk to that has been involved in a DL project?
- Do you have access to any other resource about this case that we can use to describe it?

9.3 Extended case descriptions

Case 1: Fishbrain

Fishbrain is a global social platform for people interested in fishing, founded in Sweden. It is the largest one of its kind on the market with 9 million users, and they promise anglers around the world to offer new places to fish, the ability to track and share results, as well as the opportunity to connect with like-minded users. The application launched in 2010 by the founder Jens Persson. The company employs around 100 individuals as of the interview.

Idea & pre-study

After a couple of years in service with their platform, Fishbrain realized that they had collected a unique dataset based on user-logged information including the amount of caught fish, what species, as well as other metadata such as time, date, weather, and location. The company experimented with use cases and proof-of-concepts for quite some time, but never managed to get an ML system into production. Fishbrain realized their lack of experience in developing ML, so they partnered with Modulai, a consultancy firm specialized in ML, to build a project around their data. Their idea of the data potentially having value was confirmed by Modulai:

"Fishbrain contacted us, and their proposal was special in the sense that they were well aware that they had assembled a unique dataset that was relevant and well structured." (Interview, Magnus at Modulai)

Together, they formed the idea of assisting current anglers on their platform with their fishing journey. Modulai and Fishbrain set the goal to build a ML model that predicted where a user should fish, based on what time and where they are currently located geographically. Essentially, they wanted to build a fishing forecasting model, and they chose to call it BiteTime.

As the ML case was closely linked to the organization's core business, the project group had no problem understanding the business value the project could bring. Short term, the project could lead to higher customer engagement and experience, and increasing the value of their premium in-app purchase offer. Long term, with their already large user base, leveraging their data to build a adaptive function was assumed to be a source of future competitive advantage for Fishbrain.

The feasibility of the project, however, was more uncertain. As mentioned earlier, the internal data provided by Fishbrain was well structured, but it was not certain if it was enough to work as a general recommender for anglers. Hence, Modulai began examining external data sources that could enhance the model. From research they found that weather information such as air temperature and wind speed was relevant, as well as astronomical conditions such as moon phase and solar radiation could impact fish behaviour. This type of data was easily accessible. Relevant internal and external data, as well as a clear business case, lead the parties to proceed with the project.

Planning

The team consisted of several software engineers from Fishbrain, as well as two ML engineers from Modulai. The project sponsors, Fishbrain's CTO and CEO, were both involved in the project. As Fishbrains software infrastructure was built upon Amazon Web Services (AWS), the project group decided that the overarching technology partner would be Amazon. The development of the ML model was done in python and adjacent relevant open source libraries. Their CTO reasoned with the decision as follows: *"We don't see the point of running our own infrastructure when AWS provides a more effective service that's easier to use. I want my engineers to focus on what's unique to us. Amazon SageMaker was an easy choice to help us get BiteTime into production quickly."* (AWS article)

No financial budget was made explicit, but the team specified around 4 weeks to get the first version up and running.

Execution

The data experimentation phase was accomplished in less than two months, according to the ML engineer from Modulai, and the final releasable product took 3 months to accomplish. The data consisted of over 5 million registered catches combined with geo-spatial data, which needed a certain infrastructure to be handled.

The CTO was very supportive and followed the project closely, and the team held bi-weekly updates to the whole company of their progress.

The project clearly followed Agile principles and the ML team followed the usual sprint length and ways of working at Fishbrain. Furthermore, the team closely iterated with the Fishbrain team to sanity check the results of their model. The team noticed that the model predicted sharp spikes in the probability of a certain fish catch in certain places during the day. Even though this on average would have been more accurate, the team chose to make it a little less accurate, to smooth down the probability curves, and make it fit Fishbrain's user experience and design language.

Finalization

The implementation of the model went easily, and the model is still in production in the app. The more data users log, the better the model becomes. Furthermore, as Fishbrain enters new markets, they add market-specific data and let the engineers take 1-3 days to prepare the data and finetune the model to make it more accurate.

Case 2: Non-european e-commerce platform

The case regards an non-european software company that offers an e-commerce marketplace for buyers and sellers. Sellers can set up their e-commerce store, while buyers can order and make payments. Their online services have millions of daily users. It was established twenty years ago, and today it employs several thousand people. The interview was performed with their current AI lead. The organization tried this project twice, and both of these attempts are considered in the study. The first one failed, and the latter succeeded.

Idea & pre-study

To spark customer engagement, the organization incentivizes users to leave reviews on products they have purchased in return for payments. It is encouraged, and incentivized with more money, that the review includes a picture of the purchased product. Nevertheless, they had seen some difficulties with spam as users put up images that are not relevant to the review to claim more money: *"Sellers upload goods. With this, customers can get money if they make a review. If there is an image attached they get even more reward. The problem today is that users post images that are irrelevant. We want to predict if the images that are uploaded will be relevant." (Interview with project leader)*

A failed first try

Their AI team then came with the idea to try to predict spam images with deep learning image classification. They evaluated that the business case was valid and that they had the proper data assets to give it a shot. An internal deadline of six months was set, and a project group consisting of 8 employees was put together. The team included the roles of data labelers, ML engineers and full stack developers.

Nevertheless, after nine months, the team still had no model accurately functioning. Three months overdue with no results, the project was laid off and considered a failure. According to internal research, the team spent too little time assuring the necessary quality of the data, and instead spent relatively more time, for example, in the modelling phase. Meanwhile, they knew little of the quality of their data and were all somewhat inexperienced in putting ML models in production.

A second try with a new project leader

After a new project leader had examined the previous attempt, they decided to give it a new try. The new project leader saw potential in the project as it, according to him, included low risk in deployment, and had a direct, yet moderate, business case. The spam filter could result in lower costs from not having to pay for spam, as well as customer quality from the e-commerce sellers. The project leader saw that the data and the labeling seemed pretty straightforward. In addition, building upon pre-trained models available online and then adjusted and fine-tuned for the specific use-case, the project could deliver results quickly. The project leader knew that with picture data and the task at hand, a deep learning model would provide a good tool for classification.

It was important that the model yielded a very low false positive rate, as the customer experience would suffer if correct behaviour did not lead to the cashback. The team discussed, and after experimentations and discussions with the product owner, they realized that a threshold value of around 1% false-positive classifications would have to be achieved in order for it to be fit for production.

Planning

The project leader constructed a team and he himself took on the role to lead the project, as well as being in charge of structuring data, build and train the ML model: *"The failed first attempt took months, but we had a solid estimation. I based it upon every step we needed to accomplish." (Interview with project leader)* By his side, he had one designated intern focusing on data labeling, one data engineer that assisted with the dataset handling, two colleagues assisting with integrating the API to the website, and lastly, one product owner in charge of the webpage where the ML model was to be deployed. Moreover, no budget was designated for the project, and the timeline was set to be five weeks long. With every week, focusing on a certain step of the project. The five steps consisted of labeling data, training the model, evaluating with stakeholders, develop an API, and lastly, deploying the model.

Execution

During the five scheduled weeks, the team had no scheduled meetings. Instead, they did one-to-one syncs as they stumbled upon problems with their tasks. The project leader had a clear view of the primary objective: *"Up in production as quick as possible, then you get a grasp of how it performs. That was the problem with what they had not done from the earlier project."* The team all had long experience working in an agile fashion, but they did not explicitly use any tools or procedures.

Alongside the end of the project, effective communication between the project leader and the product owner was held. The product owner was a key stakeholder as he held the final say over what goes into production or not. According to the project leader, the project saw support from management, but they were not actively involved.

The two engineers in control of building the API had no problems with deploying the model, as they had a rich experience in doing so.

Finalization

The implementation was accomplished smoothly by the team within their specified timeline. The model is currently in production and classifies thousands of images a day.

The model is so far yielding accurate results, with improvement and iteration cycles postponed or cancelled as the model accuracy is enough for the business. The product owner therefore thinks the development teams capabilities are better employed elsewhere.

Case 3: Media streaming service

This case consists of a project initiated by a media streaming service. Their offering consists of allowing content producers to upload their work and get paid by user interactions while charging the users of using their service. The organization was founded approximately ten years ago. As of this study, the organization's products offers its solution to millions of daily active users. The organization has thousands of employees. The interview was done with a former project leader in their analyst research department.

Idea & pre-study

Retention rate is a crucial metric to follow for subscription software services. It measures the percentage of users that will be active after a given period, which is essential for services that rely on either recurring revenue or ad-revenue. The streaming company was well aware of this:

We had done many similar projects before. This is one of the essential questions for our organization and other consumer related organizations." - *(Interview with project leader)*

Nevertheless, their analytics research team thought the accuracy of current human decision making on actions to increase the retention rate was overestimated, and that they probably could automate this process. This led the research analytics team to initiate this project to increase the retention rate and, therefore, recurring revenue in the future: *"We thought, let us find a more objective way to estimate this. Let us set up a bunch of features that might be relevant to see what correlates and not."*

The business case was clear. Actions to improve retention rate are revenue drivers. Moreover, large amounts of qualitative data. The problem that they saw was that although they tracked data, their decision-making process was based on human interpretation of this data.

Data sources included service usage, what client the customer was using, mobile application actions, as well as user demographics and other background variables.

Planning

The analytics research team formed a project group, consisting of three employees with data science and engineering background. The organization's overall data strategy and operations were well established, which let them not having to concentrate that much on planning API infrastructure and implementing the model. As the company had clear guidelines and processes for shipping code, deployment of models and flexible architecture, the implementation was not considered an issue. No specific financial budget was allocated, but the timeline set out was four weeks to put the model in production for the team.

Execution

The business criticality of the retention metric ensured buy-in from the company. According to the project leader, being data driven is a key component of the company culture. Therefore, he felt no issue getting support from top management and other stakeholders inside the company.

The project followed Agile principles and focused on delivering value quickly. The ML engineer mentioned the CEOs strategy of "failing fast" in order to learn quickly.

From feature engineering 40 or so features, only one was relevant: *"Then we aimed to see which features had the most predictive power if a user would be retained or not. Out all features we engineered, and took quite some time to do, we found that it was only one which was heavily correlated with if an end user would be retained."*

The finding turned the ML project on its head. With merely one feature relevant, it would be more appropriate to hard code solutions to increase retention rate, rather than to deploy their model.

Finalization

The model never went into production, yet they would never have been able to find a dependable relation without their project. According to our definition of project failure, this project was unsuccessful as it never led to implementation. Failed ML project, which lead to a successful business outcome.

Case 4: Edtech company

This case regards an AI education technology platform provider. Their offering is to let educational content providers personalize their content to its students by the case company's recommendation engine. Their business model is API request driven, and the firm was founded a couple of years ago. They are still in a startup phase with less than 50 employees, but already have thousands of users. The interview was done with their AI research engineer.

Idea & pre-study

The educational technology platform aspired to expand their current offering consisting of a recommendation engine with a new service enabled by ML. The idea was to empower the end-users to perfect their pronunciation in language learning with state-of-the-art speech recognition technology. The student records speaking out a sentence, and the model assesses the pronunciation and potential errors instantly. At the company website, the benefit of the API is described: *"The API can be used to score a word, sentence or phrase. The endpoint returns overall, word and phoneme level scores."* (The organizations website). Furthermore, the offering began as employees ideated what services that could expand the company's current offering.

Once the idea was introduced, a proof of concept was quickly created by using a Google API and training the model on some annotated speech data. The proof of concept was then evaluated with potential customers who responded very positively. From the reaction of the customers, the project continued, and they planned and started building a proper model: *"It was the sales meetings with different customers that really got us started. The reactions we received as we showed the demo. It was the demo that made the meetings very positive."*

Planning

Two developers were decided to develop the ML model, as mentioned in the interview with the AI research engineer: *"To get a week just to try to see if a model works is generally very valuable." ... "If you do not spend that much time to plan, but instead execute for a week to get up the most minimal viable solution, you get much more information on what will work and what you need to spend some time on."*

The product lead in the startup was also involved 40% to discuss the user interface of the product. Compared to their main offering, the recommendation engine, the business department was said to be involved to a less degree as the project at hand was technically challenging yet easy to understand for future customers. No financial budget was allocated, yet the two developers were decided to be given one month to have something in production.

Execution

The team performed daily stand-ups each day of the month to synchronize their work, and weekly met with with the rest of the organization. They used an online kanban project management tool to keep up with tasks that needed to be performed. For gathering the data, training the model, and deploying it, Google Cloud Platform was used, yet with no external consultants nor contacts at Google. One thing they had to tweak was the model's performance based on the students' accent, say if they were a European or Asian while learning English.

As they began deploying the model, they realized that although the training of the model was done on GPUs, the inference was still on CPUs, which in this case slowed down the API call. The team realized that this lessened the customer experience if their pronunciation was assessed slowly. Hence, running inferences on the GPU was necessary. This introduced problems with cost structure and structuring virtual machines in the cloud according to the project manager: *"It easily becomes too expensive to have many machines, and then if you run out of machines you are not able to serve requests."* For the team, this took time to solve, yet the end solution was improved significantly.

Finalization

The model was put in production and has been established as a regular offering for the edtech company. The monitoring of the model is done every month. The aim is to assess the quality of its performance in specifically chosen dimensions such as latency and number of requests per day.