Combining Data Science with Agile Software Development: A Case Study

Janne Sauvala

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 26.5.2019

Thesis supervisor:

Prof. Marjo Kauppinen

Thesis advisor:

D.Sc. (Tech.) Jaakko Särelä

Aalto University
School of Science
Author: Janne Sauvala

Title: Combining Data Science with Agile Software Development: A Case Study

Date: 26.5.2019  Language: English  Number of pages: 5+52

Software and Service Engineering

Professorship: Software Engineering

Supervisor: Prof. Marjo Kauppinen

Advisor: D.Sc. (Tech.) Jaakko Särelä

Data science has become an important topic in the context of new digital services. Many companies are creating new intelligent services that utilise data science to create better services for their users. The goal of the thesis was to investigate how companies can have successful data science projects. The case project was analysed based on 4 semi-structured interviews and end user feedback. We interviewed 2 data scientists, 2 software developers, service designer, project manager, and product director. The findings of this study were validated with an independent researcher and with the project’s data scientists.

This study revealed seven factors that affected the success of the data science project. These factors include cross-functional team, committed people, access to good quality data, elaboration of the data science system, direct and open communication, organisational support and expectation management. The data science process of the project was also an agile and iterative end-to-end process. We identified the six process activities and 15 tasks related to these activities. The activities were 1) conceptualisation, 2) problem definition, 3) data collection and preparation, 4) modelling, 5) evaluation and validation, and 6) deployment and utilisation of results.

We also identified a set of agile software development practices that can help data scientists at their work. For example, creating prototypes and iterative testing helps data scientists to experiment and validate their selected modelling techniques. Building a minimum viable product (MVP) can help data scientists and stakeholders get feedback about how end users experience digital service in practice. The results of the empirical study indicate that the data science process is an iterative, end-to-end process, where the development team is continuously learning about the data and domain. It is also important to explain to stakeholders how the data science system works and what it is capable of doing.

Keywords: data science, agile software development, success factors
Tekijä: Janne Sauvala

Työn nimi: Combining Data Science with Agile Software Development: A Case Study

Päivämäärä: 26.5.2019
Kieli: Englanti
Sivumäärä: 5+52

Software and Service Engineering
Professuuri: Software Engineering

Työn valvoja: Prof. Marjo Kauppinen

Työn ohjaaja: D.Sc. (Tech.) Jaakko Särelä


Tutkimus paljasti seitsemän hyödyllistä tekijää projektiin onnistumiselle, jotka olivat: moniosaava ryhmä, sitoutuneet tekijät, laadukkaan datan käyttö, datatiedejärjestelmän selittäminen, suora kommunikaatio, organisaation tuki ja odotustenhallinta. Projektiin datatiedeprosessi oli myös iteratiivinen ja kokonaisvaltainen prosessi. Tunnistimme 6 prosessin aktiviteettia ja 15 aktiviteetteihin liittyvää tehtävää. Aktivistetit olivat 1) konseptointi, 2) ongelman määrittäminen, 3) datan keräys ja valmistelu, 4) mallintaminen, 5) arviointi ja validointi ja 6) käyttöönotto ja tulosten hyödyntäminen.

Tunnistimme myös joukon ketteriä ohjelmistokehityksen menetelmiä, jotka voivat auttaa datatieteilijöitä heidän työssään. Esimerkiksi, prototyypiin rakentaminen ja iteratiivinen testaaminen voi auttaa datatieteilijöitä kokeilemaan ja valitsemaan heidän käyttämääen mallinnusmenetelmiä. Pienimmän toimivimman tuotteen (minimum viable product, MVP) rakentaminen voi auttaa datatieteilijöitä ja sidosryhmä keräämään palautetta loppukäyttäjiltä. Empiirisen tutkimuksen tulokset osoitavat, että datatiedeprosessi on iteratiivinen ja kokonaisvaltainen prosessi, jossa kehitystimmä oppii kokonaistenut dataasta ja sovellusalueesta. On myös tärkeää selittää sidosryhmille kuinka datatiedejärjestelmä toimii ja mihin se mahdollisesti pystyy.

Avainsanat: datatiede, ketterä ohjelmistokehitys, menestystekijät
Acknowledgements

I want to thank my supervising professor Marjo Kauppinen for guiding me throughout this thesis. My advisor, Jaakko Särelä, deserves special thanks for advising me and being invaluable support. I want to give my thanks to everyone who participated in the interviews for sharing their experiences. I want to also thank Reaktor Innovations Oy for giving me this opportunity and making this thesis possible.

Espoo, May 26, 2019

Janne Sauvala
# Contents

1 Introduction ................................................. 3  
1.1 Motivation .............................................. 3  
1.2 Research problem and questions ......................... 3  
1.3 Scope of the thesis ...................................... 4  
1.4 Structure of the thesis ................................... 4  

2 Literature review ........................................ 5  
2.1 Data science and related concepts ....................... 5  
2.2 Data science process .................................... 7  
  2.2.1 Basic data science activities ....................... 7  
  2.2.2 Activities in the big data life cycle ............... 8  
  2.2.3 CRISP-DM ........................................... 8  
  2.2.4 Data science process model ....................... 11  
2.3 Agile and lean methodology ............................ 12  
  2.3.1 Agile software development ....................... 12  
  2.3.2 Lean software development ....................... 13  
  2.3.3 Agile software development and delivery process .. 14  
  2.3.4 Agile software development practices ............ 18  
2.4 Combining data science and software development .... 19  

3 Research method .......................................... 21  
3.1 Case description .................................... 21  
3.2 Research process ..................................... 21  
3.3 Data collection ....................................... 24  
3.4 Data analysis ........................................ 25  

4 Empirical study ........................................ 27  
4.1 Overview of the case project ............................ 27  
4.2 Beneficial factors to data science project ............. 28  
4.3 Key activities and tasks of data science process ...... 31  
4.4 Agile software development practices ................ 34  
4.5 Reactions of end users ................................ 37  

5 Discussion ................................................. 40  
5.1 RQ1: beneficial factors to a data science project .... 40  
5.2 RQ2: Key activities and tasks of data science process ... 41  
5.3 RQ3: Agile software development practices ............ 43  
5.4 Limitations of the study ................................ 44  

6 Conclusions ............................................... 45  

References ............................................... 47  

Attachments ............................................... 50
1 Introduction

1.1 Motivation

Data science has been gaining popularity in the software industry and many companies have shown an increasing interest in finding ways how to benefit from data science. Companies see that data science can bring value to them and to their customers in various ways, including optimizing operations, improving decision and by making new data products (Kim et al. 2016). In a popular professional networking service, LinkedIn, the first data scientists introduced new features to the service that increased the user growth of the service. One of the new features was 'People You May Know' -connection tool that suggested new connections for LinkedIn members and it increased the growth rate of LinkedIn by significant rate (Davenport & Patil 2012).

Companies might want to use intelligent machine learning algorithms on their existing data about their users to understand customers better or implement new features to their existing products to help users to accomplish their tasks easier. There are a huge amount of possible implementations of data science and machine learning, but finding the required resources and best ways to utilise them can be challenging. The purpose of this thesis is to find out what is required from companies to have successful data science projects and how software development practices could help data scientists in achieving this.

1.2 Research problem and questions

The research problem of this thesis is:

"how companies can have successful data science projects" and the following research questions are used to find the answer to the research problem:

The research problem is divided into three separate research question that helps to find the answer to the research problem. The first research question tries to look for an overview of the beneficial factors of a data science project. The second research question focuses more closely about the data science process and tries to find its key activities. The third research question deals with concrete practices that could help the data scientists can apply in their daily work.

RQ1: What are the beneficial factors to a data science project?

RQ2: What are the key activities of the data science process?

RQ3: What agile software development practices can help data scientists in their job?


1.3 Scope of the thesis

The scope of this thesis is to focus on examining how companies can have successful data science projects. This is done by examining literature about agile software development and data science and comparing those to the case project. From literature we investigated some of the best practices and activities of agile software development and reflected them to the data science activities that we found from existing literature and from the case project. Data scientists might not have a software development background, so this thesis brings up software development practices that could help them in their job and therefore help the whole project to succeed. The findings of this thesis could be applied to similar projects that are creating digital services that are using data science to create their core value delivering features.

1.4 Structure of the thesis

The thesis starts with an introduction chapter. The introduction chapter is followed by a literature review of data science and it’s related concepts and continues on explaining agile and lean software development. In the first part of the literature review data science and it’s related concepts are explained and is followed by a description of the data science process. The second part of the literature review is an introduction to agile and lean software development methodology. Last part of the literature review contains investigation on the current state of combining data science and software development. The third chapter of this thesis describes the used research method in this thesis. The fourth chapter covers the empirical study and findings from the transcripts of the interviews. This is followed by a discussion chapter (5) where the findings are discussed more in detail. The thesis ends to the conclusion chapter (6) that contains the most important findings of this study.

Table 1 shows the chapters where the answers to the specific research questions are shown in this thesis. The table separates each research question to the literature review and to the empirical study and shows their respective chapters.

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Literature review</th>
<th>Empirical study</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1: Beneficial factors of the data science project</td>
<td></td>
<td>4.2</td>
</tr>
<tr>
<td>RQ2: Key activities and tasks of the data science process</td>
<td>2.2</td>
<td>4.3</td>
</tr>
<tr>
<td>RQ3: Agile software development practices of the data science process</td>
<td>2.3.4</td>
<td>4.4</td>
</tr>
</tbody>
</table>

Table 1: Relationship between the answers of the research questions and chapters.
2 Literature review

2.1 Data science and related concepts

Data science is a new kind of concept to describe the needs of multidisciplinary skills to analyse large amounts of complex data. People working in the field of data science needs to have good communication and presentation skills to effectively work with different stakeholders. Data scientists need to have programming skills to be able to handle big data sets with modern data science tools and advanced analytical methods to extract information from data. Data science is closely related to statistics, computer science and especially data mining (Davenport & Patil 2012, Steele et al. 2016).

Data mining, in the other hand, is “a process of discovering patterns and knowledge from large amounts of data” from different data sources, like databases, data warehouses, data streams and internet (Han et al. 2014). Leskovec et al. (2014) define data mining as discovering models for data, where the model can mean several things, like statistical models, machine learning models or computational models (summarisation, approximation or feature extraction). In the 90’s the data mining concept has been more widely used, but nowadays data science has become a more dominant concept. It seems that both of the concepts are closely related to each other.

Data mining is often used when speaking about big data. Data mining is usually done to extract useful information from very large amounts of data, big data. Wu et al. (2014) define big data as large-volume, complex, growing data sets that generate their data from several autonomous sources. These characteristics of the data make it very difficult and analyse. Wu et al. (2014) notes that getting a clear view of the data is hard because observations can be made only from a small portion of the data and the conclusions can differ a lot from each other. Analysing big data datasets requires specialised data mining platforms and algorithms, but also domain and application knowledge about the given data (Wu et al. 2014).

Data science can be used to create new kind of digital services or enhance existing services with new features. A common example of this kind of feature is recommendation engines that can recommend new or similar products for service users. Many web-shops can recommend products to customers based on their previous shopping behaviour. This kind of service needs to use machine learning to learn and recommend new products for customers. Machine learning as a computational method that uses past information (experience) to improve its performance or accuracy. The past information is usually in electric form and can be human labelled data or data that is obtained by interacting with the environment (Mohri et al. 2012). Nowadays the concepts of data science and machine learning are used as synonyms, but they mean different things, although, they can be used side by side when creating new digital and intelligent services.
Machine learning is often used with another popular computer science concept, artificial intelligence. There is no clear definition of artificial intelligence, but it describes a computational system that can act and think like a human. Russell & Norvig (2010) categorises artificial intelligence into four higher-level domains that are: thinking humanly, thinking rationally, acting humanly and acting rationally. The first two domains try to understand the intelligence of humans and try to apply it to machines. The latter two studies and applies computational models to artefacts and intelligent agents (Russell & Norvig 2010).

<table>
<thead>
<tr>
<th>Thinking Humanly</th>
<th>Thinking Rationally</th>
</tr>
</thead>
<tbody>
<tr>
<td>“The exiting new effort to make computers think... machines with minds, in the full and literal sense.” (Haugeland, 1985)</td>
<td>“The study of mental faculties through the use of computational models.” (Charniak and McDermott, 1985)</td>
</tr>
<tr>
<td>“[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning...” (Bellman, 1978)</td>
<td>“The study of the computations that make is possible to perceive, reason, and act” (Winston, 1992)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Acting Humanly</th>
<th>Acting Rationally</th>
</tr>
</thead>
<tbody>
<tr>
<td>“The art of creating machines that perform functions that require intelligence when performed by people.” (Kurzweil, 1990)</td>
<td>“Computational intelligence is the study of the design of intelligent agents.” (Poole et al., 1998)</td>
</tr>
<tr>
<td>“The study of how to make computers do things at which, at the moment, people are better.” (Rich and Knight, 1991)</td>
<td>“AI... is concerned with intelligent behaviour in artifacts.” (Nilsson, 1998)</td>
</tr>
</tbody>
</table>

Figure 1: Four different categorizations of artificial intelligence (Russell & Norvig 2010).

Nilsson (1998) adds to the definition of artificial intelligence the interaction with complex environments. This means that artificial intelligence can have a perception, ways to communicate and act in the environment. He also states that the long term goals of artificial intelligence are to create machines that can match humans or even perform better in tasks that require intelligence.

Many of the data science and its related concepts are overlapping. When domain experts and non-experts talk about data science, machine learning and artificial intelligence, they often mix concepts. The reasons for this confusion and mixing can happen because the terminology isn’t clear and there are many different competing definitions.
2.2 Data science process

The data science process is the process of how data science can be applied to solve business problems. The solution to a business problem can be an analysis of existing data, where the analysis shows before unknown relationships between different events or entities in the data, or new features to a product that gives value to the stakeholders.

2.2.1 Basic data science activities

Table 2 shows basic data science activities. The found 13 activities are separated into three different categories. The categorises are collecting, analysing, using and disseminating. In the collection phase, the data scientists were building a data collection platform that, collected data from multiple sources continuously, injected telemetry code to gather data about software execution and usage profiles and build an experimentation platform that was capable of running alternative software designs (Kim et al. 2016).

<table>
<thead>
<tr>
<th>Category</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collecting</td>
<td>Building the data collection platform</td>
</tr>
<tr>
<td></td>
<td>Injecting telemetry</td>
</tr>
<tr>
<td></td>
<td>Building the experimentation platform</td>
</tr>
<tr>
<td></td>
<td>Data merging and cleaning</td>
</tr>
<tr>
<td></td>
<td>Sampling</td>
</tr>
<tr>
<td></td>
<td>Shaping, feature selection</td>
</tr>
<tr>
<td>Analysing</td>
<td>Defining sensible metrics</td>
</tr>
<tr>
<td></td>
<td>Building predictive models</td>
</tr>
<tr>
<td></td>
<td>Defining ground truth</td>
</tr>
<tr>
<td></td>
<td>Hypothesis testing</td>
</tr>
<tr>
<td>Using and disseminating</td>
<td>Operationalizing models</td>
</tr>
<tr>
<td></td>
<td>Defining actions and triggers</td>
</tr>
<tr>
<td></td>
<td>Applying insights/models to business</td>
</tr>
</tbody>
</table>

Table 2: Data science activities (Kim et al. 2016).

In the analysis phase, the data scientists were merging and cleaning data from multiple sources (Kim et al. 2016). The data can be located in different sources and it might have missing values or imperfect values. Sampling is also one of the activities, where data scientists select a subset of the collected data and use it for their experiments. Data shaping including selecting and creating features: transforming data into new formats and creating new attributes in a feature vector. Defining
sensible metrics: defining metrics that are sensitive to data consumers. Building predictive models by using data science, machine learning and statistics. Defining ground truths: defining class labels for the data and scenarios for anomalies seen in the data. Hypothesis testing: testing new hypotheses with statistical methods.

In the use and dissemination phase, the data scientists operationalised predictive models by integrating predictive models into software products (Kim et al. 2016). They also defined actions and triggers for different prediction labels. Lastly, they translated insights and models to business value by using their domain knowledge.

It’s important to note that the data scientists might need to spend time on building the data collection platform if there is no data already collected. The time to acquire data might vary based on the collection method and how fast the data can be gathered. The analysing phase is mostly related to modifying the data and building the models. The models should become actionable and they should be used part of services and products and not just as analytic and reporting tool.

### 2.2.2 Activities in the big data life cycle

Handling and harnessing big data requires several activities. The collection, storing and analysing can be challenging tasks and requires knowledge and special tools and data pipelines. Jagadish et al. (2014) describes the steps for the usage of big data:

1. Data acquisition: data is collected from existing sources. The sources can include sensors that measure the environment and sends the data to be stored.

2. Information extraction and cleaning: the data is transformed into a better format for data analysis. In this activity, the relevant information is extracted from the data. Data cleaning is also required to fill or remove missing or faulty information from the data set.

3. Data integration, aggregation, and representation: data collected from multiple sources need to be stored in a shared place for easy access and usage.

4. Modelling and analysis: data is used for modelling and analysis. Methods include querying, mining and statistical methods of big data.

5. Interpretation: Last activity in the life cycle, where the results are interpreted by stakeholders and decisions are made based on it.

### 2.2.3 CRISP-DM

One proposed process model for data science projects is CRISP-DM (Cross-Industry Standard Process for Data Mining) (Provost & Fawcett 2013). The process model is
illustrated in Figure 2. It is a process model for the life-cycle of data mining projects and it could be used to describe the process of data scientists work. CRISP-DM is intended to work across different industries, so it can be used in various business cases and it does not depend on the data sources, data quality or the features of the data (Wirth & Hipp 2000).

Figure 2: Phases of the CRISP-DM process model (Wirth & Hipp 2000).

The CRISP-DM consists of six different phases that try to cover the whole data mining process. The phases consist of the analysis of the business problem, understanding and preparation of the existing data, modelling, evaluation of the models and deployment of the end results (Wirth & Hipp 2000). The process is meant to be an iterative process, where the process might need to start again to achieve better results, because of new knowledge that was acquired when working with the data. Some of the phases, like business understanding and data understanding, are tightly connected because the results of those phases affect highly each other. The six the phases of CRISP-DM are (Wirth & Hipp 2000):

1. Business understanding: the first phase focuses on understanding the business domain and it’s business requirements. The knowledge about data mining should be reflected in the underlying business problem and vision of how data mining would help to solve the problem.
2. Data understanding: in the second phase, the initial data collection and understanding starts. This phase gives first look to data quality problems and insights to the data and its features. Data and business understanding are in close relationship and both of them affect each other. People working on the project should reflect actively how the existing data helps them to solve business problems.

3. Data preparation: the existing data needs to be prepared for later use. The preparation can include combining data from different sources, cleaning data and transforming it into different formats.

4. Modelling: different modelling techniques are used on the data for finding a good model to solve the business problem. The data preparation and modelling can happen iteratively because different preparation methods can affect the modelling results.

5. Evaluation: the best performing models are evaluated. In this phase, it's important to check if there are any missed business requirements that the model does not cover. At the end of this phase, the results of the data mining process should be achieved.

6. Deployment: the best performing model is used to make data analysis. The results of the data analysis can be used to get insights about the data. The end result of the process could be an analysis report that can be used to solve the initial business problem.

CRISP-DM shares some similarities with previously discussed data science basic activities and activities related to the big data life cycle. Both of the described models describe activities that should be done to achieve data science goal with large amounts of data. Both of the models include the parts related to data handling, including data collection from different sources, data preparation, modelling and using the model or produced results. All of these activities are also included in the big data life cycle. The previously discussed data science activities also included more detailed tasks, like building the data collection platform and defining ground truth of hypotheses, that wasn't described in CRISP-DM -process.

CRISP-DM widens the scope of the data science process by including phases outside of the core activities related to data usage. CRISP-DM includes iterative phases of business and data understanding that are helpful not only to data scientist but to the whole organisation who own data and wants to use it to develop their business. Business understanding includes domain knowledge of the particular business and how the data can be used to solve the organisation’s business problems. CRISP-DM describes that the business understanding goes hand-to-hand with data understanding because they both relate to each other. New revelations on the business side can affect how data can be used and new ways understanding the data can lead to new ways of thinking about the business.
2.2.4 Data science process model

In a more recent study, Riungu-Kalliosaari et al. (2017) came up with another process model for data science projects. The process consists of six phases, where the whole process is iterative and covers phases for business understanding, data collection and preparation and also evaluation and deployment of the results. The phases of the process of Riungu-Kalliosaari et al. (2017) are shown in the Figure 3.

![Diagram of the data science process model](image)

Figure 3: Overall data science process (Riungu-Kalliosaari et al. 2017).

The activities in the Figure 3 process model are (Riungu-Kalliosaari et al. 2017):

1. Conceptualisation: in the first phase, the most important activity is to interact with the customer to gain an understanding of the business problem, business requirements and understand how data science could help to solve the business problem. It’s also important to manage the expectations of the customer about data science that what could be achieved.

2. Problem definition: during the problem definition phase, the business problem is formalized to an actual solvable problem and the problem is translated into a computational or mathematical problem that can be modelled. Also, identifying the customer or end users is also important in this phase.

3. Data collection and preparation: the required data is collected and prepared for the use of data scientists.

4. Modelling: after the data is collected and prepared, the modelling begins. Modelling tries to give answers to the business problem and the modelling results are usually visualized.

5. Evaluation and validation: the modelling results are evaluated and validated.
with relevant stakeholders and end users. Feedback from the stakeholders and the end users is collected and is reflected in the business problem in hand.

6. Deployment and utilisation of results: the results are utilised to solve the business problem. The results need also to be monitored. It’s also important to keep collecting feedback from the end users to see if the results stay relevant to them.

Both the data science process model and CRISP-DM are iterative and agile processes that contain the same number of phases and roughly the same phases. The data science process model emphasises that the process doesn’t just end to the deployment and utilisation of the results, but it should start a new iteration where the results are used. CRISP-DM -process doesn’t emphasise the future iterations of the process so strongly. CRISMP-DM emphasises the iterative nature of steps between business and data understanding and also data preparation and modelling. These four phases form two tightly coupled phases that need some iterative work to achieve the best results. The data science process model does not include these kinds of tightly coupled phases.

2.3 Agile and lean methodology

2.3.1 Agile software development

The roots of software methods date back to mid-1950s when iterative and incremental software development (IID) was used (Larman & Basili 2003). The methods took really off later, during the late 1990s, when Extreme Programming (XP) and Scrum were introduced (Sommerville 2016). Soon after that in 2001, the Manifesto for software development was signed by 17 people from the software industry and formed the Agile software development alliance to promote software development over plan-driven development (Fowler & Highsmith 2001, Sommerville 2016). The Agile Manifesto was written, because of the dissatisfaction to the process-heavy and plan-driven software development (Sommerville 2016). The four items of the Manifesto for software development are:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The Manifesto for agile software development says that the items on the right side have some value, but the left side items have more value. (Fowler & Highsmith 2001). There are several agile methods that guide people to develop software in a more agile way. Some research has been done studying the similarities of the agile
methods. Sommerville (2016) has found out that all the agile software development methods share the following similarities:

- Software requirements specification, design and implementation are overlapping activities. The amount of written documentation is minimised.

- The software is developed in small increments that add functionality to the software. End users and other stakeholders are involved to specify and evaluating the increments. Typically, the increments are done every few weeks.

- Development process is supported by tools that can automate parts of the software development and delivery process, like testing and deployment to end users.

Abrahamsson et al. (2002) have also done research about the common characteristics of agile methods and they found that all the agile methods are:

- Incremental (do small software releases)

- Cooperative (customer and developers work together)

- Straightforward (the method itself is easy to learn and to modify)

- Adaptive (last moment changes can be done)

2.3.2 Lean software development

During the 1940s the Japanese car-manufacturing company Toyota was still a small company compared to it’s the USA and European competitors. The other car-manufacturers had invested in mass-production lines that were able to produce cars at a cheap price. The Japanese market wasn’t big enough to copy the mass-production lines, so Toyota needed to find alternative ways to make their cars. (Poppendieck & Poppendieck 2003)

During that time in Toyota, the lean method started to evolve. In the mid-1980s, the term 'lean' was first used in the context of the production management process and product development to describe the production methods of Toyota. By using the lean principle, Toyota was able to achieve the same results in about half of the labour hours as their USA and European car manufacture competitors. (Poppendieck & Cusumano 2012).

The lean development consists of seven principles that effect to the whole development organization and they are (Poppendieck & Poppendieck 2003):

1. Eliminate waste: waste is everything that does not bring value to the customers. In software development, the waste can be features that are not used by the end users or written documentation that is not read by anyone. Also, waste is also created when development is handed off from a team to another. The way how to minimize waste is to identify exactly what the customers want and then develop and deliver that.
2. Amplify learning: developing software is also a learning process for the development team. The lean method encourages the whole team to keep learning constantly more about the customers and the product.

3. Decide as late as possible: decision making should be delayed as much as possible. By making decision later time, the development team have had more time to gather information and are more capable to make effective decisions.

4. Deliver as fast as possible: by delivering software products faster, the team can learn faster by gathering users feedback. Rapid development cycles allow teams to gather information faster and make better decisions.

5. Empower the team: the development team should have the power and responsibility to make an important decision about the product they are developing. The development team has first-hand experience and the best knowledge about the product. If the decision making is centralised to one or a few persons who are not working closely to the product, then a lot of information is lost and the time to make a decision is going to increase.

6. Build integrity in: the software has coherent integrity in architecture, usability and extensibility. High integrity cannot be built by processes or procedures, but with wise leadership, effective communication, relevant expertise and healthy discipline.

7. See the whole: it is important to see the big picture of the software product and not focus on small different parts. With too much of focusing on different parts, it’s easy to forget aspects that affect the whole product, like the overall performance of the product.

The principles of lean manufacturing were later introduced to software development. Table 3 shows how the lean principles were adapted to Microsoft’s software development practices.

### 2.3.3 Agile software development and delivery process

Sommerville (2016) identifies that agile software development consists of three main activities: specification, development and validation. During the specification, the outline requirements of the software are discovered. In agile software development, use-cases and user stories are a popular way to formalize the requirements from the end user’s perspective. During the development, the actual implementation of the software happens and the needed features are added to the product. After the required implementation is done, it is validated against the specification. It is important to find possible faults or errors during the validation and make sure the software is doing what it’s supposed to do. Cao & Ramesh (2008) found out that end users and other stakeholders should be part of the software validation because their feedback can help to find problems in the software product.
<table>
<thead>
<tr>
<th>Toyota-style “lean” production (manual demand-pull with Kanban cards)</th>
<th>1990s Microsoft-style “agile” development (daily builds with evolving features)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIT (just-in-time) “small lot” production</td>
<td>Development by small-scale features</td>
</tr>
<tr>
<td>Minimal in-process inventories</td>
<td>Short cycles and milestone intervals</td>
</tr>
<tr>
<td>Geographic concentration - production</td>
<td>Geographic concentration - development</td>
</tr>
<tr>
<td>Production leveling</td>
<td>Scheduling by features and milestones</td>
</tr>
<tr>
<td>Rapid setup</td>
<td>Automated build tools and quick tests</td>
</tr>
<tr>
<td>Machine and line rationalization</td>
<td>Focus on small, multifunctional teams</td>
</tr>
<tr>
<td>Work standardization</td>
<td>Design, coding and testing standards</td>
</tr>
<tr>
<td>Foolproof automation devices</td>
<td>Build and continuous integration testing</td>
</tr>
<tr>
<td>Multiskilled workers</td>
<td>Overlapping responsibilities</td>
</tr>
<tr>
<td>Selective use of automation</td>
<td>Computer-aided tools, but no code generators</td>
</tr>
<tr>
<td>Continuous improvement</td>
<td>Postmortems, process evaluation</td>
</tr>
</tbody>
</table>

Table 3: Comparison between the lean production of Toyota and the agile development in Microsoft (Poppendieck & Cusumano 2012).

In agile software development, the activities are done as parallel activities, not as clearly separated activities. The parallelisation allows the developers to discover hidden requirements or errors during the development phase while doing the development and validation at the same time.

Agile software development is usually done in small increments that add functionality to the software product. The outline description of the software lays down the initial requirements for the software and the software is developed iteratively in small increments until the software full fills the outline description and is ready and the final version is ready to be used by end users. Sommerville (2016) This incremental development process is modelled in Figure 4.

Sommerville (2016) has gathered three different benefits of incremental development, compared to a waterfall software development model:

1. The cost of changing software requirements is decreased because the amount of analysis and changes is documentation is reduced.

2. Feedback from customers and end users can be collected in a much earlier phase. This is because the small increments can be showed to the customer
Figure 4: Incremental development process (Sommerville 2016).

and they can comment them better than just software specifications.

3. The software can be developed and delivered to the customer much faster. The delivered software might not have all the features implemented, but customers can start to use it with a smaller set of features.

The incremental software development has also some negative effects on project management. Because of the much less amount of created documentation, the development process is not so visible to the project management. Also, because of the small increments and rapid changes, the architecture of the software tends to degrade. The development team needs to allocate time regularly to refactor the internal structure of the software. (Sommerville 2016)

Incremental development has also lead software engineers to think about delivering the software in small releases to customers. This method is called incremental delivery and it emphasizes building the software product in small increments and delivering the product faster to customers (Sommerville 2016). The delivered product does not have all the planned features implemented, but the customer does not need to wait that the product has all the features ready but can start to use it sooner with fewer features. The incremental delivery process is shown in Figure 5.
Figure 5: Incremental delivery process (Sommerville 2016).

Sommerville (2016) found out that incremental delivery process has several benefits:

1. Customers can start using the product much earlier phase and gain experience about the new features. Customers can also give early feedback and help the developers to enhance the product.

2. Customers do not have to wait until the whole product is ready, but they can start using it when the absolutely necessary features are complete and gain value from the product.

3. The incremental delivery process ensures that small increments to the product should be easy to make.

4. Because the most necessary features are prioritized and developed first, they are validated the most. This leads to decrease the number of software failures in the most important parts of the system.

Incremental delivery has also some drawbacks that are (Sommerville 2016):

1. Because the requirements are not formalized in very detailed level, it can be hard to identify the most critical parts of the system that are common to all
the parts of the system.

2. If the developed software is replacing a legacy system, maintaining all the features to be used by the users can be hard. The new system does not have all the features of the old system, so it makes it difficult for the customers to switch to use the new system.

3. The iterative process can be hard to accommodate for companies and organisations that require a formal and complete specification of the system.

Incremental development and delivery are good methods for some software projects. For very large scale projects and projects where there are several distributed teams, the incremental development method might not be the best option. Also, projects that have critical safety or security requirements and therefore need more detailed specification, the incremental model is not good. (Sommerville 2016)

2.3.4 Agile software development practices

Agile software development brought new concepts and work practices to software development when it started to become popular in the late 1990s. Most of the practices came from agile Extreme Programming, Scrum and Crystal methodologies (Begel & Nagappan 2007, Fitzgerald et al. 2006). This analysis also included two studies related to agile requirements engineering (Cao & Ramesh 2008, Bjarnason et al. 2011). Table 4 contains common used agile practices used in the software industry.

The practices have been categorised into four categories: planning, development, testing and ways of work. The **planning category** includes practices that are used usually before actual feature implementations have started. Planning also guides development work in a longer perspective, like what features should be implemented and when they should be released based on business priorities and technical estimates (Fitzgerald et al. 2006, Bjarnason et al. 2011). The scope of the development sprint should follow one extremely prioritised backlog list (Cao & Ramesh 2008, Bjarnason et al. 2011).

Common **development practices** focus on creating features iteratively and sharing knowledge inside the team. Many of the practices emphasise the teamwork, team-wide practices and collective ownership.

**Testing practices** contains practices related on agile testing, where new code is continuously integrated to the rest of the software and tested against the user stories and needs with automated tests (Begel & Nagappan 2007, Cao & Ramesh 2008, Fitzgerald et al. 2006).

**Ways of work** includes practices that are common to the whole agile software development process. In an agile context, direct communication, cross-functional teams and frequent updates are an important role.
<table>
<thead>
<tr>
<th><strong>Category</strong></th>
<th><strong>Practice</strong></th>
<th><strong>Description</strong></th>
<th><strong>References</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Planning game</td>
<td>Quick determination of the scope of the new release</td>
<td>(Bjarneason et al. 2011) (Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Extreme prioritisation</td>
<td>Keep the product backlog always prioritised</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Constant planning</td>
<td>Change and add requirements when needed</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Small releases</td>
<td>Iterate in small releases</td>
<td>(Bagel and Nagappan 2007)</td>
</tr>
<tr>
<td>Development</td>
<td>Prototyping</td>
<td>Create fast prototypes to test implementations</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Pair programming</td>
<td>Developers use one computer to do programming in pairs</td>
<td>(Bagel and Nagappan 2007) (Bjarneason et al. 2011)</td>
</tr>
<tr>
<td></td>
<td>Simple design</td>
<td>Use simple system design to solve the problem</td>
<td>(Bagel and Nagappan 2007) (Bjarneason et al. 2011)</td>
</tr>
<tr>
<td></td>
<td>Collective code ownership</td>
<td>Share knowledge of the code inside the whole team</td>
<td>(Bagel and Nagappan 2007) (Bjarneason et al. 2011)</td>
</tr>
<tr>
<td></td>
<td>Team coding standards</td>
<td>Share similar coding style with the whole team, including variable and function naming conventions</td>
<td>(Bagel and Nagappan 2007) (Bjarneason et al. 2011)</td>
</tr>
<tr>
<td></td>
<td>Iterative and integrated requirements engineering</td>
<td>Requirements engineering is done iteratively as the development progresses along with other development activities</td>
<td>(Cao and Ramesh 2008) (Fitzgerald et al. 2006)</td>
</tr>
<tr>
<td></td>
<td>Sustainable pace</td>
<td>Do development in sustainable and constant speed</td>
<td>(Bagel and Nagappan 2007)</td>
</tr>
<tr>
<td>Testing</td>
<td>Continuous integration</td>
<td>Integrate new changes early with the rest of the code base and test it with automated tests</td>
<td>(Bagel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Reviews and acceptance tests</td>
<td>Testing the software against user stories, needs and requirements</td>
<td>(Bagel and Nagappan 2007) (Cao and Ramesh 2008) (Fitzgerald et al. 2006)</td>
</tr>
<tr>
<td></td>
<td>Test-driven development</td>
<td>Developers write unit tests first and then the actual feature against the tests</td>
<td>(Bagel and Nagappan 2007) (Bjarneason et al. 2011) (Cao and Ramesh 2008)</td>
</tr>
<tr>
<td>Ways of work</td>
<td>Direct interaction with customer</td>
<td>Do not separate software developers from customers</td>
<td>(Bagel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Cross-functional teams</td>
<td>Team should have all the skills they need to build the product</td>
<td>(Fitzgerald et al. 2006)</td>
</tr>
<tr>
<td></td>
<td>Face-to-face communication</td>
<td>Favour face-to-face communication over other communication methods</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Daily stand-up meeting</td>
<td>Short meeting with the whole team to keep everyone updated on the progress</td>
<td>(Bagel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>System metaphor</td>
<td>Common understanding with the customers and developer how the system works</td>
<td>(Bagel and Nagappan 2007) (Bjarneason et al. 2011)</td>
</tr>
<tr>
<td></td>
<td>Design improvement</td>
<td>Don’t focus on getting design correct on the first try, but improve it as you go with refactoring</td>
<td>(Bagel and Nagappan 2007) (Bjarneason et al. 2011)</td>
</tr>
</tbody>
</table>

Table 4: Agile software development practices.

Agile software development is not done in a step-by-step way. This means that many of the practices described in this chapter doesn’t happen strictly in their specified category, but they can happen in parallel. For example, prototyping is a practice that could be used as part of planning, development or testing depending on the purpose of the prototype. Prototyping in planning could be used to check the feasibility of new technology. In development, it could help to see if new technology could be integrated into the existing software. As a part of testing, prototypes could help to test if a new kind of user interface could fulfil user needs.

### 2.4 Combining data science and software development

Software analytics has been used in the software industry to gather information about the software development process or the usage and performance of the software product. Menzies & Zimmermann (2013) describes software analytics as analytics
for empowering individuals and teams to gain and share insights from their data to make better decisions. The data used for analytical purposes can be gathered from several places. The data can be gathered from the software development processes to understand defects or the software development process in general or the data can be gathered from the end users software to see how they are actually using the software (Menzies & Zimmermann 2013, 2018). Some of the use cases of software analytics use cases are described in the Table 5.

<table>
<thead>
<tr>
<th>Use case of software analytics</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understand software development process inside the organisation</td>
<td>Track software defects and development progress. Make estimates based on the progression data.</td>
<td>Menzies and Zimmermann (2013)</td>
</tr>
<tr>
<td>Understanding how software defects happened and how to prevent them</td>
<td>Use analytics data to find the root causes of defects. Use data to prevent similar defects by e.g. reducing software complexity</td>
<td>Menzies and Zimmermann (2018)</td>
</tr>
<tr>
<td>Detect anomalies and malfunctions in large software systems</td>
<td>Run performance tests on large software systems and gather analytics logs. Anomalies, like high power consumption, can be detected and predicted from patterns.</td>
<td>Zhang et al. (2013)</td>
</tr>
<tr>
<td>Get insight how end-users use software product</td>
<td>Data scientists can use software analytics data to model end-users behaviour about how they use the software product. This behaviour can be used detect if some of the features are not used.</td>
<td>Kim et al. (2016)</td>
</tr>
</tbody>
</table>

Table 5: Software analytics use cases.

Some of the use cases of software analytics are commonly used methods and have been used effectively in the industry. Defect detection and system analysis are common practices nowadays and lately the methods have been improving by the coming of data science (Menzies & Zimmermann 2013, Zhang et al. 2013). There clearly is a trend on the value of data science in software development and how data science can help organisations to create even better digital products by making data-driven decisions and helping organisations to create value to users (Kim et al. 2016, Menzies & Zimmermann 2018).

This study does not cover data science in the software development area in the context of software analytics. This study focuses on data scientists, who are part of the team creating new services that use data science, machine learning and artificial intelligence to create new kind of digital services that disrupt existing markets.
3 Research method

3.1 Case description

Reaktor Innovations Oy is the case company for this research. Reaktor Innovations Oy is a consultant company that has been established in 2000. Reaktor started as a small firm but has grown to a company of over 500 employees in five different locations (Helsinki, Amsterdam, Dubai, Tokyo and New York). The core capabilities of Reaktor includes engineering, product and service design, business design, communication design, organization design and future design.

The core strategy of Reaktor is to work closely with clients and combine multidisciplinary teams that can iterate and deliver high-quality products and services fast. Some of the most notable customers include Airbus Defence and Space, Down Jones, Finnair, HBO, Kone, Nasdaq and Supercell.

One of Reaktor’s recent data science projects was chosen for this research as a case study. The project was designed and developed by Reaktor with the customer organisation between the spring 2016 and spring 2017. The end product was a web application that used data science to estimate the sale price of apartments.

3.2 Research process

The study was done using a case study method on one of the data science projects done by the case company. Case study is a good qualitative method for developing a detailed and deep understanding of the studied concept (Silverman 2005) (Rune-son & Høst 2009). The case study method can also reveal how and why certain phenomenons occur and explain the cause-effect relationship between these phenomenons (Easterbrook et al. 2008). Because of the limited amount of available data science projects to study, the chosen project needed to be understood well in details. Qualitative research method provides a good way to truly understand technical and human complexities well in details (Seaman 1999).

The empirical research process is based on qualitative research interview process (King 2004) and interview study procedure (Karlsson et al. 2007). The whole research process of this case study is visualized in the Figure 6.

The qualitative interviews were constructed and carried out in four steps: define interview questions, create interview guide, recruit interview participants and carry out interviews (King 2004). The interviews itself in this study were constructed in four steps: interview planning, data collection, material analysis and content analysis (Karlsson et al. 2007).

The interview questions were done by creating the first set of questions and then modifying them in four iterations. The questions were evaluated in every iteration by other experienced researcher and once by a third external researcher, who didn’t
participate to the research in other way, at the end of creation process. The first version of the questions was created by using the research questions as the starting point and by applying the writer's own knowledge about software engineering and data science.

Interview guide was created to give an introduction to the interviewees. The guide
included a structure of the interview and topics that needed to cover during the interviews. These topics included introduction of the interviewers, introduction to the interview and research, confidential statements, brief overview of the question topics and ending words of the interview.

All of the interviewee participants were members of the case project. They were recruited by recommendations from the key members of the project team. We wanted to interview members with different backgrounds, so we could get a good view of the multidisciplinary team and understand the project in a whole and from different angles. There were several people working on the project, including twelve people from Reaktor and client’s project director, client’s executive team and legal department. Some team members of the development team from Reaktor didn’t work on the project from start to finish. For example, team members focusing on concept creation did most of their work on the initial steps of project and left the project when the actual software development work started. The team composition and the structure of the project is illustrated in the Figure 7.

![Diagram](image)

**Legend**
- **Client organisation**
- **Case company**
- **CONCEPT CREATION (2)**
- **Communication channel. Thickness of the arrow describes the activity of the communication**
- **Separate tasks and how many persons were working on that**

**Figure 7**: Team composition and structure of the project.

During the initial concept creation phase of the project, there was no project manager in the team, but the concept creators met with the client and showed their initial concepts of the service. Later on the project a person who took the role of the project manager joined to team from the case company. Project managers role
was primary to communicate with the product director and present teams results
to the executive team. Project manager also invited team members to demo the
service to the meetings of the execute team. The development team also needed
legal department’s help to clarify some legal concerns that was raised during the
project.

The most key persons of the case project participants were interviewed. The key
persons included the two data scientists of the project, two longest staying software
developers, the main concept creator, the project manager and the project direc-
tor. The project director was the only representative of the client organisation and
rest of the interviewed participants were from the development team of the case
organisation.

3.3 Data collection

There are three different kinds of qualitative interview methods; structured, un-
structured or semi-structured and, group interviews. In the structured interview
the interview has preplanned and complete set of interview questions. Only the
planned questions are asked and the answers are recorded. In the unstructured or
semi-structured interview, the interviewer has an incomplete set of questions and he
may ask other additional questions (Myers & Newman 2007). Flick (2010) points
out that in this interviewing method, the planned questions should be more or less
open-ended and the interviewer should let the interviewee to answer to them freely.
This way the interview questions work as a guiding tool for the interview. In the
group interview setting, there are a small group of interviewees and the interview is
more or less structured. The interviewer need to guide the discussion in the inter-
view, so all the participants can bring up their own views and that way the whole
group can give their answers to the questions. (Flick 2010)

The interview conducted in the research were semi-structured single-person or group
interviews and were audio recorded by a mobile phone. The interview questions were
acting as a guiding tool for the interview and possible follow-up questions were asked
if the interviewee answer needed more explanation to clarify the meaning of words
or to clarify why something happened in the project.

The first interview was conducted by a single researcher and rest of the interviews
by two researchers. The first interview had one interviewee and the rest of the
interviews were organised as group interviews of two to four participants. The case
organisation and client organisation interview participants were interviewed in their
organisations own premises. In total, seven members from the case company and
one from the client organisation was interviewed in four separate interview sessions
during the spring 2018.
Table 6: Interview participants.

<table>
<thead>
<tr>
<th>Organisation</th>
<th>Role</th>
<th>Years working in the company</th>
<th>Total experience in years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case company</td>
<td>Project manager</td>
<td>1.5</td>
<td>10+</td>
</tr>
<tr>
<td></td>
<td>Conceptualisation</td>
<td>2.5</td>
<td>20+</td>
</tr>
<tr>
<td></td>
<td>Data scientist</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Data scientist</td>
<td>&lt; 1</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Software developer</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Software developer</td>
<td>2</td>
<td>19+</td>
</tr>
<tr>
<td>Client</td>
<td>Product director</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

3.4 Data analysis

The basis of the qualitative data analysis has three stages when analysing a big and complex phenomenon (Seaman 1999):

1. Analysis. By analysing and identifying the researcher identifies separate components inside the studied phenomenon or substance.

2. Study. The found components are studied carefully to get a deep understanding of each component.

3. Understanding. The gathered knowledge about each component is then brought together and this knowledge is then used to get a better understanding of the original phenomenon that consists of the smaller components.

In this study, the stages of the qualitative data analysis were followed by using the grounded theory. Grounded theory was first introduced by Glaser and Strauss in 1967 (Seaman 1999). The data analysis process had five stages. The first stage was transcription the audio files of the interviews to written forms. The rest four stages followed the grounded theory analysis process:

1. Open coding. Analysing the interview transcripts carefully and finding interesting facts by marking the interesting parts with codes. These codes were used later to develop concepts.

2. Development of concepts. The found coded facts from the transcript are grouped together to form similar concepts. These concepts were related to products development, teamwork, data science etc. Coded facts were not restricted to be part only one concept but they could be part of multiple different concepts.
3. Grouping concepts into categories. Categories formed higher level topics that shared a common theme. Themes were used to draw conclusions about the case project. The conclusions could include observations about the teamwork, data modelling, software development activities, challenges the team faced or beneficial factors to the success of the project.

Seaman (1999) points out that the grounded theory has several good advantages but also several drawbacks. One of the clear advantages is that the formed theories are backed up by the data. The method is also very straightforward and it encourages the researcher to study the data early on. For novice researchers, the coding of the data might be overwhelming and the researcher is also subject to personal biases when doing data coding and formalising theories. Personal biases might affect to what facts are coded and how the facts are used to formalise to concepts, categories and theories.

The found themes were used to formalise answers to the research questions. One of these concrete themes formalised this way was how important explanation of the AI system to the stakeholders is. Both of the data scientists and several other team members came up several times in the interviews that they had to spend time explaining how the prediction model worked to stakeholders. Data scientists needed to demonstrate and explain how the prediction model worked to the steering group and had to work with copywriters to explain the service to end users.

The found results of this study were first validated with another researcher who was attending to the interviews and with another independent researcher who wasn’t part of the interviews. After the first validation, the results were presented and validated with the two data scientists who worked in the case project. Finally, the results are presented in this thesis.
4 Empirical study

4.1 Overview of the case project

The most important events of the project are visualized in the timeline of the Figure 8. In spring 2015 the customer organisation started a development program that was looking for new ways to improve their business in their home services-sector. At the start of the development program, the client organisation came up with several ideas about how to start to improve their business. The organisation had to choose the most promising idea to focus on and start to develop the idea further. The chosen idea to be developed was the automated tool for estimating house prices, because it had a promising idea to become a bigger marketplace for new services and it could funnel more customers to their existing services.

![Timeline of the project](image)

Figure 8: Timeline of the project.

Later in 2015 some initial service creation and design were created by another consultant company before Reaktor joined to the project in April 2016. Reaktor quickly took the main responsibility of the business and service design. Reaktor was also responsible for the implementation of the service - including designing the underlying machine learning algorithms also implementing the website interface that was shown to consumers.

During fall 2016, the client’s steering group of the project showed dissatisfaction towards the look and feel of the user interface and wanted it to be changed. Also, the accuracy of the prediction raised some concerns, because they thought that the accuracy should be higher to be useful for their customers and better for the brand of the organisation. During the September 2016, several team members were switched in the team and the project continued with new team members.

In November 2016, the interviewed team members felt that the service was good enough to be released. They had modified the design and improved the accuracy of the prediction model based on the client’s feedback. The client planned the launch to be done during spring 2017 with media coverage. During the first weeks of the
launch, the service had several tens of thousands of unique visitors and after the launch some thousands of visitors each month.

At the end of the project, the client decided to move the project to another team that was maintaining services related to the domain of housing services. Reaktor’s team did changes to the service, created documentation and handed over the product to the other team. On Reaktor’s behalf, the project ended to this transition.

4.2 Beneficial factors to data science project

One of the goals of the empirical study was to identify beneficial factors for a data science project. Based on the data gathered from the interviews, we identified seven different beneficial factors: cross-functional team, committed people, access to good quality data, explanation of the AI system, direct and open communication, organisational support and expectation management. We further analysed the data and factors and grouped the identified factors to four higher-level categories: team, data science, communication and organisation (Table 7).

<table>
<thead>
<tr>
<th>Category</th>
<th>Factor</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team</td>
<td>Cross-functional team</td>
<td>• Service development requires skills from many areas: data science, user-centric design, concept creation, software development, testing and validation.</td>
</tr>
<tr>
<td></td>
<td>Committed people</td>
<td>• The project might face big challenges and commitment helps the team to overcome them.</td>
</tr>
<tr>
<td>Data science</td>
<td>Access to good quality data</td>
<td>• Helps creating an accurate prediction model.</td>
</tr>
<tr>
<td></td>
<td>Explanation of the AI system</td>
<td>• There might be roadblocks, like legal concerns, that prevents the usage of data.</td>
</tr>
<tr>
<td>Communication</td>
<td>Direct and open communication</td>
<td>• Exploration of the prediction model to stakeholders and what features affects to it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Discuss about the possibilities and weaknesses of the data and data science with stakeholders.</td>
</tr>
<tr>
<td>Organisation</td>
<td>Organisational support</td>
<td>• Gather feedback from all the important stakeholders.</td>
</tr>
<tr>
<td></td>
<td>Expectation management</td>
<td>• Discuss about concerns and possible challenges before they become roadblocks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read the stakeholders’ nonverbal signals.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Support from the organisation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Approval of all the important stakeholders.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Share the same vision about the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Discuss about the possibilities and weaknesses of the data, prediction model and the service.</td>
</tr>
</tbody>
</table>

Table 7: Beneficial factors to data science project.

Cross-functional team: The interviewees felt the forming a cross-functional team was an important part of the project’s success. They said that the project had a wider skill variety than their previous projects. The team needed to cover skills on digital service conceptualisation, data science, software development, user interface and interaction design, content writing, user testing and validation.
At the start of the project, the business and service designers surveyed customers the hardest or most frustrating tasks related to selling or buying a house. The customers told that the most difficult part of the process was to get a realistic estimation of an apartment’s price. After finding this pain point in the customer’s journey, the team started to conceptualise a solution to this problem. They invited the first data scientist to join the project. Together with the client, they started to look for a way to solve the problem and they found a usable data that could be used to build a service, where customers could estimate the prices for apartments.

A lot of work was done related to content writing. The team thought that it was very important to deliver the message of the service to the customers clearly and in that way that they would understand how the prices of the apartments were estimated. The customers should understand the concept without knowing the mathematics and statistics behind it. Together with data scientists, the content writers created the texts to carry a message that could be understood by common people. The content writers validated the content within the team and with the client.

The team also validated the accuracy of the prediction model with the client. The client was very strict on the accuracy of the estimation because they thought that a poor estimation would be bad marking. They wanted that the customers could get reliable and accurate numbers so that the estimation would be helpful to the customers.

The development team also needed help from the domain and legal experts of the customer organisation, because they did not have these skills inside of their team.

**Committed people:** The interviewees told that commitment was an important factor for the success of the project. Especially, the project manager of the team showed grit to push the project forward by getting the necessary resources for the team and negotiating with different stakeholders. There is a high chance that this kind of new and complex projects will face challenges related to the access to good quality data, building the prediction model or organisational challenges. The development team faced an unseen challenge that their initial data was not enough to get accurate predictions and they needed to get access to a better data source. The team finally got permission to use the data, but they needed support from the organisation to pass legal obstacles.

**Access to good quality data:** The team quickly iterated with their first data set to build a prediction model to estimated the apartment prices in the biggest cities in Finland. However, after seeing the results the client wanted to extend the service to cover more cities. Also, the client wanted the prediction accuracy to be better because they felt that 10 per cent accuracy wasn’t enough and the data scientists should get it down to 5 per cent. The data scientists needed to explain to the client that the data they have in use cannot be used to cover more cities and increase the prediction quality. The quality and coverage of cities in the data was a limiting factor.

The client decided that it was crucial for the service and it’s business to increase
the coverage and accuracy. The team with the help of client discovered a better
data source for their use. Unfortunately, there were legal limitations about who has
access to the data and use it. The legal department of the client helped the team to
get access to the data by handling the legal issues related to the data usage.

**Explanation of the AI system:** The data scientists told that they needed to
elaborate on how the data science model works to various stakeholders. The stake-
holders included the steering group, other team members and end users. They also
needed to discuss the possible weaknesses and possibilities with the steering group,
so they could come up with a good solution together. One of the examples that they
came up together was to add a description to users how the data science model does
the prediction and how different features affect the results. Because the users were
not experts in mathematics or in data science, it was also important that the added
description was written in a form that was easy to understand.

**Direct and open communication:** The development team felt that they didn’t
receive all the necessary feedback. At the start of the development work, the team
didn’t have a direct communication channel to the steering group and only the
product director was representing them in the meetings with the steering group.
The team felt that if somebody else was representing them, they could hear only
“bits and pieces” from the conversations from the meetings.

After the project manager from the case company joined the project, the team had
a direct communication channel to the client organisation’s steering group. The
team members were able to have a better discussion about the project and also
read the important nonverbal signals of the steering group. The team also formed
direct relationships with the important decisions makers of the steering group, so
they could have conversations outside of the formal meetings. The development
team and project director felt that the formal meetings were too short to discuss all
the details and some of the discussion must have to be held outside of the formal
meetings.

Project manager emphasised the importance of open communication. By openly
communicating about their concerns and wishes, the team and the client were able
to get answers to difficult questions before they become major problems. One of
the examples was that the steering group told their concerns about the overall lay-
out of the service and the error margin of the data science model. Based on the
given feedback, the team was able to do the necessary steps to improve the service.
The development team also communicated openly about their concerns towards the
steering group, so they could get help.

**Organisational support:** The development team faced obstacles related to access
to better quality and more complete data. Without the organisational support from
the top-level management, the team might not have access to this data and the
created model would have not been so accurate and complete. The top-level man-
agement saw the importance of this project and got the organisation’s own legal
department to help the team to get access to the better data source.
The customer organisation had several stakeholders, who needed to evaluate the service before it could be published. The most important stakeholder was the top-level managers of the steering group. The steering group included the highest-ranking managers of the housing sector and they were overseeing the service and its development. The client’s legal department was also a key stakeholder by making sure that the new service had all the legal obligations done right. Lastly, the marking department was also interested in the project. In the interviews, the team members mentioned that they needed to get approval from client’s marketing department to publish to service because the marketing department needed to prepare a press release and marketing material for the launch. The marketing also made sure that the upcoming launch didn’t interfere with other press releases of the company and therefore not getting enough attention from media.

**Expectation management:** The data scientists told that they needed to communicate, visualise and discuss the possibilities and limitations of the data to the steering group. The steering group was excited about the possibilities of the service and wanted include more uses cases to the service, but the team recommended to leave them out of the project’s scope because the amount of work would have increased dramatically. The team also tried to keep the scope of the project smaller, so that they could launch the new digital service as soon as possible. The team felt that they should launch the service publicly, so they could learn more about the users’ needs and develop the service further based on the feedback.

The concept designers wanted to first build an MVP (minimum viable product) of the product first, with a limited feature set. The MVP would have been used to test how users reacted to the service and use the feedback from them to improve the service. However, one of the concept designers told that he felt that they were not on the same page with the steering group about the scope of the project. This led to problems because the team was building a service with a minimal set of features and the steering group was expecting a service with all features implemented.

### 4.3 Key activities and tasks of data science process

One of the goals of the empirical study was to identify the key activities and tasks of the data science process. In this chapter, we present the six activities and 15 tasks related to these activities. The activities are first presented by Riungu-Kalliosaari et al. (2017) in their research done in the exact same company as this research’s case company. The six activities are conceptualisation, problem definition, data collection and preparation, modelling, evaluation and validation and deployment and utilisation of results (Figure 9). The categorisation of activities led to the categorisation of the found tasks.

In the first activity, conceptualisation, the service designers identified the potential customer’s use cases and recognised customer’s needs that could be fulfilled with a new service they could build. During this activity, the designers noticed that the
new service could use data science to fulfil the customer’s needs.

In the second activity, problem definition, the team started to work with one of the data scientists to formalise the problem into a form that could be solved with data science methods. The main activities included investigating if they have the necessary data, can the data be used for modelling and could the prediction model help to solve the customer’s problem.

During the third activity, data collection and preparation, the client provided a way to the team to access the data. The data collection process in this case project was quite easy because the data was given to the team and they didn’t need to start collecting the data by themselves. One of the initial steps of the data scientists was to prepare the data to be used to generate a prediction model. The data preparation included modifying the data format to a format that could be used more easily in software the data scientists were using to build the prediction model. The data scientists also needed to check if the data had inconsistent, missing or otherwise bad data points that needed to be removed from the data set.

In the fourth activity, modelling activity, the data scientists were building the prediction model that would be used in the new digital service. The performance of the model was critical because the prediction accuracy was one of the core requirements
<table>
<thead>
<tr>
<th>Category</th>
<th>Practice</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Planning game</td>
<td>Quick determination of the scope of the new release</td>
<td>(Bjarnason et al. 2011)</td>
</tr>
<tr>
<td></td>
<td>Extreme prioritisation</td>
<td>Keep the product backlog always prioritised</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Constant planning</td>
<td>Change and add requirements when needed</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Small releases</td>
<td>Iterate in small releases</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td>Development</td>
<td>Prototyping</td>
<td>Create fast prototypes to test implementations</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Pair programming</td>
<td>Developers use one computer to do programming in pairs</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Simple design</td>
<td>Use simple system design to solve the problem</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Collective code ownership</td>
<td>Share knowledge of the code inside the whole team</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Team coding standards</td>
<td>Share similar coding style with the whole team, including variable and function naming conventions</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Iterative and integrated requirements engineering</td>
<td>Requirements engineering is done iteratively as the development progresses along with other development activities</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Sustainable pace</td>
<td>Do development in sustainable and constant speed</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td>Testing</td>
<td>Continuous integration</td>
<td>Integrate new changes early with the rest of the code base and test it with automated tests</td>
<td>Begel and Nagappan 2007</td>
</tr>
<tr>
<td></td>
<td>Reviews and acceptance tests</td>
<td>Testing the software against user stories, needs and requirements</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Test-driven development</td>
<td>Developers write unit tests first and then the actual feature against the tests</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Direct interaction with customer</td>
<td>Do not separate software developers from customers</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Cross-functional teams</td>
<td>Team should have all the skills they need to build the product</td>
<td>Fitzgerald et al. 2006</td>
</tr>
<tr>
<td></td>
<td>Face-to-face communication</td>
<td>Favour face-to-face communication over other communication methods</td>
<td>(Cao and Ramesh 2008)</td>
</tr>
<tr>
<td></td>
<td>Daily stand-up meetings</td>
<td>Short meeting with the whole team to keep everyone updated on the progress</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>System metaphor</td>
<td>Common understanding with the customers and developer how the system works</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
<tr>
<td></td>
<td>Design improvement</td>
<td>Don’t focus on getting design correct on the first try, but improve it as you go with refactoring</td>
<td>(Begel and Nagappan 2007)</td>
</tr>
</tbody>
</table>

Table 8: Key activities and tasks of data science process.

in the service and the results of the prediction model would be shown directly to the customer. The data scientists worked iteratively, trying different approaches to building the model to make the best prediction with the lowest error.

The main purpose of the evaluation and validation activity was to test the prediction model and the new service. The data scientists tested the model against the data they had got from the client in earlier activities and also evaluated it with stakeholders. The data scientists created prototypes and used them to gather feedback from stakeholders. Early prototypes included drawings and digital prototypes, but later stage prototypes were done via a web-interface. The development team tested the prototypes with the members of the client organisation. The data scientists also presented their work to the steering group directly. The data scientists needed to explain and show to the client organisation how the prediction model worked. Most of the feedback the team received in this activity, was concerning about the UI/UX of the service and the error margin of the prediction model. The steering group wasn’t happy with the user interface and layout of the service and they gave
feedback that the error margin of ten per cent was too high and they wanted it to be five per cent. Most of the evaluation done in this activity was done internally inside the development team and the client organisation and not with real end users. The rising concern related to the high error margin made the development team look for another data source because the current data could not be used to create a more accurate model. The client organisation helped the team to get access to better data, but it raised new legal problems that the client organisation had to solve before the data could be used. This shows, how committed the client organisation was to this project because they were willing to use more legal resources to create the service.

In the last activity, deployment and utilisation of results, the service was made public to everyone. The client organisation made a press release about the new service and marketed it in social media. Web analytic about the behaviour of the visitors on the website was acquired and it showed that most of the initial hypotheses of the potential target customer segments were correct. Other user behaviour analytics in the service was collected from the users and the team saw that some of the features of the service weren’t used - only the two core features was the most used. The team noted that they could have just focused on these features and leave the rest of the features out or for later implementation. Feedback was also collected from the users. Users were able to give feedback in many ways, including feedback forms with closed- and open-ended questions and open feedback forms. After this activity, the project was ending and the team needed to prepare a handover to another team, who would be responsible for the maintenance of the service. Before the handover, the team analysed the gathered analytic and feedback and made a list of suggestions on how to improve the service and presented it to the client organisation.

4.4 Agile software development practices

One of the goals of this study was to find out if there are any agile software development practices that could be helpful for data scientists. In this chapter, we present the seven practices that we found. The found seven practices were grouped into three different categories that represent different aspects of the project’s work. The practises, categories and their explanations are shown in the Table 9 followed by a more detailed explanation of the different practices.

Iterative development and testing. By developing the product in small iterations, the results of the iteration can be shown to the customer and end users to confirm if the developed features are something they want to have. Small iterations also minimise the possibly wasted work, because the committed work-effort stays relatively small. Wasted work can happen when the development team misunderstands the requirements of the system and develop something unnecessary that has to be discarded.

Prototyping and visualisation. It is a good practice to separate the data science
<table>
<thead>
<tr>
<th>Category</th>
<th>Practice</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>Iterative development and testing</td>
<td>- Confirm what the customer wants.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Keep “wasted” work minimal.</td>
</tr>
<tr>
<td></td>
<td>Prototyping and visualisation</td>
<td>- Separate data science module from the actual implementation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Build interface between data science module and rest of the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Faster experimentation and testing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Share easily with stakeholders.</td>
</tr>
<tr>
<td></td>
<td>MVP (minimum viable product)</td>
<td>- Agree with stakeholders what MVP means in the project</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Collect feedback from the usage of MVP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Validate the service with users.</td>
</tr>
<tr>
<td>Team</td>
<td>Cross-functional teams</td>
<td>- Cover more different skill areas.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Project needs more skills than just mathematical and programming skills.</td>
</tr>
<tr>
<td></td>
<td>Shared working space</td>
<td>- Faster communication with team members.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Helps to have better discussion about complex topics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Reuse the material in the working space.</td>
</tr>
<tr>
<td></td>
<td>Pairworking</td>
<td>- Knowledge sharing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Helps to validate ideas.</td>
</tr>
<tr>
<td>Communication</td>
<td>Direct communication channels</td>
<td>- Data scientists need to explain the behaviour of the prediction model to different stakeholders.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Easier to have discussion on complex topics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Helps data scientists to get the resources they need.</td>
</tr>
</tbody>
</table>

Table 9: Practices helping data scientists.

module from the actual implementation and add an interface that connects the data science module to the rest of the service. By this separation, it’s easier and faster to do changes to the model and still iterate on the interface, without it letting to affect the data science model.

Creating prototypes helps the data scientists to experiment and validate their selected modelling techniques. Prototypes also work as a visualisation to stakeholders, so they can see how the actual service might look and work. Stakeholders can more easily understand and discuss an actual prototype than abstract concepts. This also could shift the focus of the stakeholders to details of the prototype. When focusing on the details it is possible to lose the sight of the whole service concepts and make changes to much less important aspects of the service.

**MVP (minimum viable product).** Building an MVP helps the data scientists and stakeholders see what kind of service they are building in practice. The MVP can be used to validate an idea, collect feedback from stakeholders and confirm requirements. In the case project, the development team wanted to create an MVP to see how the service will be seen by the end users. Because the development team and steering group didn’t have a common understanding, what an MVP means, the
development team built some features that did not add much value to the end users. For example, one of the features was a web-based chat that enabled the end user to ask questions from the real-estate agent of the customer organisation. However, only a few questions were sent through the chat.

**Cross-functional teams.** The project that aims to create a new digital service might need to cover a lot of different skills, from software development to design and content writing. It is a good practice to have most of the needed skills inside the team, so the team can work autonomously and progress the project without too many dependencies outside of the team. For example, the project needed more skills than just skills related to data science. Data scientists might need help from service designers and software developers. The interviewed data scientists didn’t handle the user interface design, content writing or most of the software development. The other work related outside of data science domain was done by other team members.

**Shared working space.** Using the same shared working space lowers the barrier to have conversations and makes communications faster, compared to emails or messages. Data science includes complex topics that can be hard to explain with traditional communication methods, like emails, and can be explained much easier face-to-face.

The team can reuse the space and material if they can use the same space and they don’t need to relocate to a new space. Saving your ideas on the walls of the room can help on communication by storing the information in a very easily accessible location where everyone can see it.

**Pairworking.** One of the data scientists told that they have had discussions in the company to place always two data scientists or data scientist and data engineer together in a project. Pair working is a good way to share knowledge with other members of the team. With shared knowledge, it also makes easier to change members in the team, when all of the knowledge is carried by just one of the team members.

One of the data scientists raised concerns about the validation of the used data science technique and possible mistakes: “It is not easy to know, if it cannot be done with the given data, is the chosen model wrong or if there is a bug (in the code)?”. This kind of challenges could be avoided by letting someone else check the data scientist’s results.

**Direct communication channels.** Direct communication with stakeholders make it easier to have a discussion about complex data science topics. The data scientists needed to discuss about the prediction model and the underlying data with several stakeholders, including the managers of the customer organisation and legal experts. Without this direct communication, it might have been difficult to understand the concepts of data science and what can be achieved with the data.

The data scientists convinced the stakeholders of the customer organisation to get access to a better quality data source. The data scientists created a demo that
showed how much better results they could get by using the better data source and this was enough for the managers to get the access to the data by using customer’s own legal experts.

4.5 Reactions of end users

The client organisation posted advertisements for the launched service to social media. The advertisements were meant to get users to visit to use the service to get a price estimation of their own house and also provide feedback. We analysed the reactions and comments that end users posted to the advertisement. The advertisements were posted around the same time as the public launch happened and several after the initial launch and those advertisements gathered over 400 comments from end users. The advertisement posts and end user comments were publicly available to all the users of that specific social media platform.

We categorized the end user’s comments to five different categories, depending on the content of the comment: **satisfied with results, improvement suggestions, questions or confusion about the service, scepticism towards the service or service provider, and not satisfied with the results.**

**Satisfied with results.** Most of the end users who posted comments to the advertisement were satisfied with the service. The satisfied end users were happy with the price estimation they received because it was accurate or the estimation was higher than they initially paid for the apartment. They also had knowledge about the general house prices in their area, so they could compare the estimated price to the actual market value. They also realised that the estimation algorithm isn’t perfect and it might not always work accurately.

**Improvement suggestions.** Some of the end users had improvement suggestions. They noticed that with small modifications the service could be more helpful for them. Some of the modifications could have not been possible to implement, because the used data was missing the needed features. One of these missing features was data about the view of the apartment from the windows and another one was if the apartment was the last house in terraced house building. These kinds of properties increase the price of the apartment, but that information was not stored in the data, so the data scientists couldn’t take it into account in the model.

**Questions or confusion about the service.** Some end users had questions or they were confused about the service. Users were wondering, why the service couldn’t predict their house prices or why the estimated price was lower than they expected. The service was build only to predict prices of block apartments, but some users tried to use it to estimated prices of real estate apartments. Estimation with lower price than the user has bought the house left the users wondering why the price has
dropped suddenly from what they have paid for their house a few years ago - this made the users fear if they have paid too much from their apartment. Also, some of the users were wondering if the client organisation is going to replace their own real estate agency with the digital service.

**Scepticism towards the service or service provider.** Some users raised the question about how reliable a machine’s prediction can be, compared to a professional real estate agent’s experience. The customer service answered to these questions that a professional real estate agent should make the final evaluation of the price because just using the location data of the apartments cannot give an exact prediction. Some users were concerned about the motivations of the service provider. The users were asking if the service provider is estimating lower prices than market value in purpose, so the service provider could take advantage of the cheaper price and sell it to their customers who invest to apartments.

**Not satisfied with the results.** Dissatisfied users commented that the estimation was much lower than they were expecting. Some users understood that the service provides only estimations and there is a change it goes wrong. In some cases where the price estimation was low, it could be explained by unique features of the apartment, like pipe renovation is done and the condition of the building. These features were not stored in the data, so it could have not been modelled.

Clearly, the service provided help with accurate estimation to most of the end users. It is also understandable that the service couldn’t predict the prices well in edge cases that had a unique view from the apartment or had some features that weren’t stored in the data. In these cases, a professional real-estate agent could make better estimation and help the end users to sell or buy an apartment with a market value price. More concerning reactions from the users were those, where the user feared if the real estate service was going to be replaced by digital service or if the service provider has some hidden agenda. Some of the users were concerned if they have to start using the digital service for now on and not be able to get a human real-estate agent to help them. The end user concerns about the service providers hidden agenda could be bad for branding. Those comments are available for everyone to see and could damage the brand.

These customer comments revealed new views to us how to evaluate intelligent digital services. When building intelligent digital services, you need also to consider how you communicate about how it’s going affects to your existing services. These new services could replace manual human work or it could co-exist as supporting service by helping users to get what they need fast and later direct the users to contact the experienced professionals. For some, replacing human work can be a negative approach, because that would mean that some would lose their job because of software. Another important aspect is about the explanation of intelligent systems. If you cannot tell the users clearly how the system is working, the users can become suspicious towards it. The gathered comments of end users revealed how important
it's to think widely about the concept of intelligent service by identifying all the stakeholders and their possible reactions to it.
5 Discussion

5.1 RQ1: beneficial factors to a data science project

We identified seven beneficial factors to a successful data science project that were: cross-functional team, committed people, access to good quality data, elaboration of the AI system, direct and open communication, organisational support and expectation management. These factors were divided into four different categories: team, data science, communication and organisation.

Beneficial factors related to the success of the team was that the team was cross-functional and the team members were committed to the project. The goal of the project was to build a new digital service that used data science as its core function. In addition to data science the project needed a lot of other skills, like concept creation, project management, software development, copy-writing and design skills. When a company is building a new service using data science, the project also needs the same resources and skills needed for a more traditional software development project.

One of the beneficial factors related to data science was access to good quality data. Without good quality data, building a good prediction model is impossible and the built service wouldn’t serve it’s a customer in a good way. In this project, the access to the data was challenging, because the client and the development team were sure that they have access to the data they can use, but later in the project it revealed to them that there were legal restrictions about using the data. With the help of the client’s legal department, the team was able to use the data for their modelling purposes.

The other beneficial factor related to data science was the validation of the AI system. It was important that stakeholders of the project understood how the system worked and agreed that is works in the correct way. Even the end users needed to have a good explanation about how the systems worked. Lack of good explanation might affect the impressions of end users. End users could see the service in a negative light because they feel that the system does not produce reliable results based on their experience and viewpoint. Other stakeholders additional to end users has to have insight into how the system works. The decision makers might have their own vision about how the AI should work, but without knowledge of the limitations of the system the service might not meet the stakeholders’ expectations.

Communication is an important factor in project work and we found the same results in the case project also. Direct and open communication enables the team and client to discuss openly possible concerns and challenges before they escalate to a roadblock for the project. The team also needed to gather feedback from all the important stakeholders. Without a direct communication channel to the stakeholders, some of the important knowledge would have been lost, because the team would have not been able to read e.g. the stakeholders’ non-verbal signals.
The whole organisation needs to support the ongoing data science project. The project might encounter hard obstacles that need to be dealt with before releasing the service. In the case project the team had to deal with problems related to data quality and the model’s estimation procession. However the organisation was willing to provide higher quality data with better coverage of different house prices. During the project some legal questions were raised but the organisation’s lawyers were able to help the project to solve the legal concerns. Without the support starting from the steering group to other parts of the organisation the project might have not to be as successfully or might not have been released at all. These examples show how important the whole organisation is in supporting data science projects that are built around new services and processes.

5.2 RQ2: Key activities and tasks of data science process

In the literature review we described the basic data science activities (Kim et al. 2016) and two data science process models (Provost & Fawcett 2013, Ruungu-Kalliossaari et al. 2017). The data science activities were categorised to three categories of collecting, analysing and using and disseminating. Some of the most important activities found by (Kim et al. 2016) were:

- Collecting: building the data collection platform
- Analysing: data merging and cleaning, building predictive models
- Using and disseminating: hypothesis testing, operationalizing models, applying insights/models to business

These activities were picked because they were identified in the case project or because they are a prerequisite for the other activities to happen. The building the data collection platform -activity is important because without data no data science can be done. In this case project, the team didn’t need to build the data collection platform because they got the data straight from the client organisation.

The data scientists told that they needed to do data cleaning. They didn’t need to do data merging maybe because they had only one source of the data and it didn’t need to be combined with any other sources. Building the predictive model is the key activity for any data scientists if they have to create a system that can do prediction. In the case project, the service was predicting house prices based on various attributes of the house, including location, size, year of construction and many others.

The case project’s team did hypothesis testing by validating the created predictive model and by validating the service concept with the client’s steering group. The team also operationalized the model by creating an online service that used the model. The client organisation also started to utilise the results of the project (feedback from the steering group and end users) in other projects and develop the prediction model further.
These basic activities are an essential part of the data science process. All of the described activities are presented in the two found data science process models. The process models don’t only describe the activities but also describe their order and relations in phases.

The two studied process models shared some similarities and have some differences. Almost all of the described basic data science activities from the study of Kim et al. (2016) were found in the two process models, but the activities contained some of the more detailed activities that the process models didn’t include.

The two data science process models contained both six major phases. The main phases of CRISP-DM (Cross-Industry Standard Process for Data Mining) -process was business understanding, data understanding, data preparation, modelling, evaluation and deployment (Provost & Fawcett 2013). CRISP-DM is a process model for the life-cycle of data mining projects and it could be used to describe the process of data scientists work.

The data science process model published by Riungu-Kalliosaari et al. (2017) included conceptualisation, problem definition, data collection and preparation, modelling, evaluation and validation, deployment and utilisation of results. The found activities of Riungu-Kalliosaari et al. (2017) were presented in a case study done by in the same company as this research’s case company.

Both of the described process models are agile, meaning that the activities are done iteratively and incrementally. The processes are also an end-to-end process, where the process starts from the initial conceptualisation of the service and ends to the deployment of the service. The processes encourage the participants to learn continuously about the data, model, business and users of the service. Without continuous learning, the first developed approach might not work and lead to a failing product when it does not meet the expectations of the users.

In this case project, the process model for data science projects done by Riungu-Kalliosaari et al. (2017) might fit better to describe the phases of the data science phases. The whole project got an idea from the client organisation when they started to create a concept for a new digital service that would use their existing data. The people working on conceptualisation found an opportunity in the market that a tool for house price estimation could improve the organisation’s presence in the market. After the initial concept, the data scientists joined to project and started to define the problem in a mathematical way that could be solved by using data science. The data scientists created iteratively the model for house price estimation and the rest of the web-service was developed iteratively side-by-side the model. Finally, the results of the service satisfied the client organisation’s steering group and the service was launched publicly. The feedback from end users was collected and the results lead to new ideas for the next steps on how to develop the service. The case project followed the data science process model described by Riungu-Kalliosaari et al. (2017). However, some parts of the process might be done in smaller iterations, so that the whole project doesn’t need to be deployed or released. In CRISP-DM there are four
phases that are tightly connected by iterative loops forming two bigger phases. The four phases are business and data understanding and data preparation and modelling. These four phases form two tightly coupled phases that need some iterative work to achieve the best results.

Both of the process models are meant to be iterative. This means, that the process might need to start again to achieve better results, because of new knowledge that was acquired when working with the data. Sharma et al. (2017) also notes the importance of an agile method for data mining. They noticed that it is hard to come up with planned specific objectives before looking at the available data and resources first. Only after acquiring some basic level of knowledge about the data and the business domain, can people working with the data get something meaningful aspects out of the data.

5.3 RQ3: Agile software development practices

Some of the widely used agile software development practices can be helpful for data scientists. For the development work, iterative development and testing, prototyping and visualisation are recommended. These practices help the data scientists to communicate easier with different stakeholders and confirm that their expectations are met. Also, developing an MVP (minimum viable product) version of the service can be helpful, because the feedback collection from users and customers can be started sooner than waiting for ready service that has all of the features implemented.

In the field of software engineering, prototyping works well to elicitation and validation of software requirements and it can be used to explore different software solutions and user interface design (Sommerville 2016, Kapyaho & Kauppinen Aug 2015). These same principles can work in the context of data science when stakeholders are evaluating the data science models or the user interface that is using the underlying model. While prototyping could help stakeholders to communicate their requirements, prototyping could also bring a new set of challenges. Cao & Ramesh (2008) found out that prototypes can easily become production code because stakeholders assume prototypes to be working software and they do not accept longer development times. This can lead to a situation where stakeholders have higher expectations about the software and the situation of the project when they see working prototypes but do not realise that there is still much work to be done to get the software ready for end users (Kapyaho & Kauppinen Aug 2015). The interviewed team members also noted that the steering group started to comment more about the outlook of the service rather than focusing on more critical issues. This is one example of how prototypes can lead the discussion on wrong tracks when stakeholders can see a working prototype.

The team structure shared working space, pair-working and communication came up as important practices. The importance of a cross-functional team that works in the same physical space was highly annotated. It allowed the same team to
mostly design and implement the whole service by themselves. The data scientists
of the development team also throw an idea about pair-working in the data science
project. The effectiveness of pair-working has been studied in software engineering
and Williams et al. (2000) have reported good results in the productiveness, software
quality and developers satisfaction to work. In the famous Extreme Programming
-method, both the pair-programming and using an open-workspace are thought as
the major practices of Extreme Programming (Beck 1999).

5.4 Limitations of the study

This study takes into account only one data science project. Projects that create
new digital services can vary from each other in terms of their progress and outcomes.
The project is also heavily affected by the composition of the team, individual team
members, the scope of the project and the client organisation. Most of these variables
are human factors and they might or might not appear in similar projects. Also, not
all of the team members were interviewed, but only the most relevant to the success
of the project. Some other team members might have different views about the
phases of the project and might answer differently to the interviewing questions.
From the client organisation, only the responsible project manager was interviewed
and not stakeholders who were guiding the creation of the digital service.

A small sample of interviews and only one case project does not guarantee data
saturation. On qualitative research, data saturation is an important aspect that is
required to achieve reliable research results. Fusch & Ness (2015) defines that data
saturation is reached when there is a way to obtain new information and the obtained
information does not add anything new to the already found material. This study
used only a small number of interviews from one case project. Our findings might
differ from other projects that have used data science and therefore more research
would be needed to make the results more reproducible.

Interviews as a data gathering method have its own drawbacks and weaknesses
because there are several factors that might go wrong. These are things like that
the interview feels too artificial to the interviewee or the interviewer lacks trust
towards the interviewer (Myers & Newman 2007). The interview questions were
systematically reviewed by two experienced researchers. By reviewing, we tried to
guarantee the neutrality of the questions and keep the questions not leading the
interviewees’ answers.

The used data analysis method, grounded theory, can also produce wrong results
depending on the bias of the researcher. The researcher might read the collected
data too much from his viewpoints and then lead to wrong theories. (Lazar et al.
2017) The data analysis was mainly done by only one researcher. Letting multiple
researchers participate in the coding and analysis of the data, would have increased
the reliability of the findings and possibly uncovered new findings.
6 Conclusions

This thesis focused on researching on how companies can have successful data science projects. In this thesis, we investigated one project whose goal was to create a new digital service to estimate apartment prices by using data science as its core feature. The created digital service was used by consumers and allowed them to see the price estimation of the apartment based on previous transactions.

The results of the study suggest that many of the basic agile practices of a software project are also useful in data science projects. This means that the modern agile software development practices can be used in the data science project to have good results. Modern software development practices include having a cross-functional team that can work closely together and communicate directly and openly with all the important stakeholders. This way of working decreases the risks of the project falling into trouble or building something that stakeholders do not see valuable. If the project seems to face challenges, it is important that the rest of the organisation is willing and capable to support the team.

The results from the empirical study indicate that the data science can add an extra layer of challenges to the existing software development challenges. The new challenges are related to data: Can the customer’s problem be solved based on available data and how to get access to good quality data that can be used for this purpose? These data science related challenges need to be tackled during the software development and these challenges might not be trivial to solve. It might need a lot of discussion with different stakeholders and support from the organisation to overcome these challenges.

The results of the empirical study also indicate that the data science process is an iterative, end-to-end process, where the team is continuously learning about the data and domain. Setting goals to data science projects can be challenging, if the availability of the modelling data is uncertain or if the quality or the coverage of the data is not good enough. By building the service iteratively, the team can validate that it is possible to build the wanted features with the given data. The iterative results can also be presented to stakeholders in forms of prototypes or minimum viable products, in this way, feedback can be gathered earlier.

The results of the empirical study also point out that it is important to explain how the data science system works and what it is capable of to stakeholders. The data scientists need to be able to communicate clearly how the prediction model works. This information needs to be presented clearly to customers and end users because lack of knowledge about the mechanics why they receive certain results might confuse or mislead them. We saw in the findings that the results the customer did not expect from the digital service confused them and made them even suspicious towards the service and the brand. It is also important to explain to stakeholders what can be done with data science and what it is capable with the given data. Otherwise, the stakeholders’ expectations might grow too high and they
feel disappointed when the service does not meet their expectations.

In this study, we found out that the end users' reactions to the service were mixed. Most of the reactions were positive when the users got results that were aligned with their expectations. But in the case the price prediction from the service was different what they expected, the reaction towards the service and brand was more negative. To reduce the negative reactions and emotions towards the service would require more studies on how to explain the system’s behaviour to non-technical users who have a basic knowledge of the application domain. These recommendations and predictions about the apartment prices could also affect the lives of the customers financially if they would use the prices that the system recommended. It would be interesting to study, what would be the best practices to explain how an AI systems works.

Another interesting research topic came up during the interviews: the validation and proving the correctness of an AI and prediction system. In more complex cases, where an AI would decide if a bank should give a loan to an applicant or how a patient should be treated in a hospital. These questions are hard and complex and in the case of medical treatment decisions are done today by a doctor. In the future, these hard tasks might be tried to automate by an AI system and the validation of the results of the AI system is especially critical because then there are bigger stakes and possibly human lives in risk.
References


ment - an interview study with practitioners’, Information and Software Technology 49(6), 588–604.


King, N. (2004), Using Interviews in Qualitative Studies IN Cassell and Symon (Eds.) Essential guide to qualitative methods in organizational research, 1. publ. edn, Sage Publications Ltd.


### Questions for product owner and the development team

<table>
<thead>
<tr>
<th>Question number</th>
<th>Question position</th>
<th>RQ focus</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Background introduction</td>
<td></td>
<td>Could you briefly tell about your background: How long have you worked in this company and in this project? What kind of education you have?</td>
</tr>
<tr>
<td>2</td>
<td><strong>Important</strong></td>
<td></td>
<td>Could you tell your role and responsibilities in the project?</td>
</tr>
<tr>
<td>3</td>
<td><strong>Can be skipped</strong></td>
<td></td>
<td>How long experience do you have from your current role?</td>
</tr>
<tr>
<td>4</td>
<td>Start of the project</td>
<td>RQ2</td>
<td>How did this project start?</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>RQ2</td>
<td>When did this project start and what is the current situation?</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>RQ1</td>
<td>Where did you get the idea of leveraging data science/machine learning in the project?</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>RQ1</td>
<td>What was the goal of the project in the beginning and has it changed (did it change during the project)?</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>RQ3</td>
<td>What kind of phases have you had in the project?</td>
</tr>
<tr>
<td>9</td>
<td>Product development</td>
<td>RQ1</td>
<td>Who has participated in the project and what have they done?</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>RQ3</td>
<td>Based on your experiences from the project, which activities and good practices related to the development work would you recommend?</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>RQ3 (If answered to question 10)</td>
<td>Does your work approaches and practices, related to the development work, in this project differ from your previous ones on similar projects? How they differ?</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>RQ1</td>
<td>Have you used data science/machine learning to build some other products or services before? If you have, briefly tell me more about those products and services.</td>
</tr>
<tr>
<td>13</td>
<td>Ending and future plans</td>
<td>RQ1</td>
<td>What do you think about the results of the project?</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>RQ1</td>
<td>Do you know who have used the results of the project and how?</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>RQ1</td>
<td>What has been crucial for the success of the project?</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>RQ1</td>
<td>Is it somehow possible to measure the success of the project?</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>RQ1</td>
<td>Which things have gone well in the project and why?</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>RQ1</td>
<td>What kind of problems or challenges have you had in this project?</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>RQ1</td>
<td>Could you think any ways to solve or ease those problems and challenges?</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>RQ1</td>
<td>What is the goal of the project now?</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>RQ1</td>
<td>What are the future plans for the project?</td>
</tr>
<tr>
<td>Kysymyksen numero</td>
<td>Kysymyksen sijoitus</td>
<td>RQ kohdistus</td>
<td>Kysymys</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>1</td>
<td>Taustatiedot</td>
<td>RQ2</td>
<td>Kuinka tämä projekti sai alkunsa?</td>
</tr>
<tr>
<td>2</td>
<td>Tärkeää</td>
<td>RQ1</td>
<td>Mistä saitte idan hyödyttää &quot;datatiedettä&quot;/koneoppimista projektissa?</td>
</tr>
<tr>
<td>3</td>
<td>Voi ollaan</td>
<td>RQ2</td>
<td>Kuinka paljon sinulla on aiempaa kokemusta nykyisistä työtehtävistäsi?</td>
</tr>
<tr>
<td>4</td>
<td>Projekti aloitus</td>
<td>RQ2</td>
<td>Milloin projekti alkoji ja mikä on sen nykytä?</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>RQ1</td>
<td>Mikä oli projekti tavoitte alussa ja onko se muuttunut (muuttui) projektiin aikana?</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>RQ3</td>
<td>Millaisia vaiheita projektissa on ollut?</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>Tuotekehitys RQ1</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>Projekti kokemusten pohjalta, mitä hyviä toimintatapoja ja käytäntöjä liittyen kehitystyöhyön suosittelisitte muille vastaavanlaisille projekteille? RQ3</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>(Jos vastasi kysymyksen 10) Eroavatko teidän tässä projektiassa käytännöine toimintatavat tai käytännöt teidän aiemmissa samankaltaisissa projekteissa käytännöistä kehitystyöhyön liittyvistä toimintatavoista tai käynnistä? Kuinka ne eroavat? RQ3</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>Olleetko käytänteet aiemmin &quot;datatiedettä&quot;/koneoppimista tuotteiden tai palveluiden rakentamisessa? Jos olette, niin kertoile lukeutu syseistä tuotteista tai palveluista. RQ1</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>Lopetus ja tulevaisuuden suunnitelmat RQ1</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td>Tiedätkö ketkä ovat käytänteet projektin tuloksia ja millä tavalla? RQ1</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td>Mikä on olut ratkaisevaa projektiin onnistumisen kannalta? RQ1</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td>Onko jotkin mahdollista mitata projektiin onnistumista? RQ1</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td>Mitkä asiat ovat menneet hyvin projektissa ja mikä? RQ1</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td>Millaisia ongelmia tai haasteita tellä on ollut projektin aikana? RQ1</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td>Tuleeko mieleenne tapoja ratkaista tai lieventää kyseisiä ongelmia ja haasteita? RQ1</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td>Mikä on projektin tavoitte nyt? RQ1</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td>Mitkä ovat projektin tulevaisuuden suunnitelmat? RQ1</td>
</tr>
</tbody>
</table>