

UNIVERSITY OF LJUBLJANA
SCHOOL OF ECONOMICS AND BUSINESS

MASTER'S THESIS

**BUSINESS VALUE ESTIMATION OF SOFTWARE
REQUIREMENTS BY PRODUCT OWNERS IN THE PROCESS OF
SOFTWARE DEVELOPMENT**

Ljubljana, June 2026

TIANA GROSHEVA

AUTHORSHIP STATEMENT

The undersigned Tiana Grosheva, a student at the University of Ljubljana, School of Economics and Business, (hereafter: UL SEB), author of this written final work of studies with the title “Business value estimation of software requirements by product owners in the process of software development”, prepared in collaboration with mentor Associate Professor Dr. Anton Manfreda.

DECLARE

1. this written final work of studies to be based on the results of my own research;
2. the printed form of this written final work of studies to be identical to its electronic form;
3. the text of this written final work of studies to be language-edited and technically in adherence with the UL SEB Technical Guidelines for Written Works, which means that I cited and / or quoted works and opinions of other authors in this written final work of studies in accordance with the UL SEB Technical Guidelines for Written Works;
4. to be aware of the fact that plagiarism (in written or graphical form) is a criminal offence and can be prosecuted in accordance with the Criminal Code of the Republic of Slovenia;
5. to be aware of the consequences a proven plagiarism charge based on the this written final work could have for my status at the UL SEB in accordance with the relevant UL SEB Rules;
6. to have obtained all the necessary permits to use the data and works of other authors which are (in written or graphical form) referred to in this written final work of studies and to have clearly marked them;
7. to have acted in accordance with ethical principles during the preparation of this written final work of studies and to have, where necessary, obtained permission of the Ethics Committee;
8. my consent to use the electronic form of this written final work of studies for the detection of content similarity with other written works, using similarity detection software that is connected with the UL SEB Study Information System;
9. to transfer to the University of Ljubljana free of charge, non-exclusively, geographically and time-wise unlimited the right of saving this written final work of studies in the electronic form, the right of its reproduction, as well as the right of making this written final work of studies available to the public on the World Wide Web via the Repository of the University of Ljubljana;
10. my consent to publication of my personal data that are included in this written final work of studies and in this declaration, when this written final work of studies is published;
11. that I have verified the authenticity of the information derived from the records using artificial intelligence tools.

Ljubljana, June 28th, 2026

Author's signature:



ABSTRACT

This study explores whether and how product owners estimate the business value of software requirements so they can effectively prioritise their product backlog in the process of software development. It additionally focuses on the definition of business value and the factors and methods product owners consider when prioritising software requirements. The research consisted of a literature review and empirical qualitative research, where eight product owners with different backgrounds were interviewed using semi-structured interviews, and observations were conducted as a supplementary method. The findings reveal that many product owners rarely use structured business value estimation techniques to understand the business value of software requirements and tend to base their prioritisation decisions on various factors, on intuition, or rely on other decision-makers. The results contribute to a set of business value estimation techniques and prioritisation methods which can be used in practice. Overall, this study contributes to understanding that product owners need more clarity around the concept of business value and the estimation techniques that they can use to prioritise more effectively.

KEY WORDS: business value, prioritisation methods, estimation of business value, product owner responsibilities, business value of software requirements

SUSTAINABLE DEVELOPMENT GOALS



POVZETEK

V magistrskem delu proučujem, kako produktni lastniki ocenjujejo poslovno vrednost programskih zahtev za učinkovito prednostno razvrščanje. Dodatno se osredotočam na opredelitev poslovne vrednosti ter na dejavnike in metode, ki jih produktni lastniki upoštevajo pri prednostnem razvrščanju programskih zahtev. Raziskava temelji na pregledu literature in empirični kvalitativni raziskavi. V empiričnem delu sem izvedla osem intervjujev s produktnimi lastniki z različnimi strokovnimi ozadji. Kot dopolnilno metodo sem uporabila tudi opazovanje. Ugotavljam, da številni produktni lastniki redko uporabljajo strukturirane tehnike za ocenjevanje poslovne vrednosti programskih zahtev. Pri odločitvah o prednostnem razvrščanju se pogosto opirajo na različne dejavnike in intuicijo ali pa se zanašajo na druge odločevalce. Rezultati prispevajo k naboru tehnik za ocenjevanje poslovne vrednosti in metod prednostnega razvrščanja, ki jih je mogoče uporabiti v praksi. Na splošno z magistrskim delom prispevam k razumevanju, da produktni lastniki potrebujejo jasnejše

razumevanje koncepta poslovne vrednosti in tehnik ocenjevanja, ki jim lahko pomagajo pri učinkovitejšem prednostnem razvrščanju programskih zahtev.

KLJUČNE BESEDE: poslovna vrednost, metode prednostnega razvrščanja, ocenjevanje poslovne vrednosti, odgovornosti produktnih lastnikov, poslovna vrednost programskih zahtev

CILJI TRAJNOSTNEGA RAZVOJA KLJUČNE BESEDE

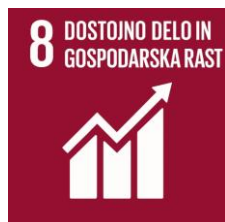


TABLE OF CONTENTS

1	INTRODUCTION.....	1
2	AGILE SOFTWARE DEVELOPMENT.....	3
2.1	The transition from traditional to Agile software development.....	4
2.2	Delivering value as the main focus of Agile software development.....	6
2.3	Agile methodologies and frameworks	7
2.4	The Scrum framework	8
3	THE ROLE OF PRODUCT OWNER IN SOFTWARE DEVELOPMENT	10
3.1	Distinction between the product owner and product manager	11
3.2	Main challenges of the product owner	11
3.3	Understanding software requirements	13
3.3.1	Software requirements elicitation.....	14
3.3.2	Software requirements prioritisation	15
4	BUSINESS VALUE ESTIMATION OF SOFTWARE REQUIREMENTS	17
4.1	Definition of business value.....	19
4.2	The purpose and techniques of business value estimation	22
4.2.1	Quantitative techniques of estimating business value	24
4.2.2	Other techniques of estimating business value.....	25
5	EMPIRICAL RESEARCH ON BUSINESS VALUE ESTIMATION OF SOFTWARE REQUIREMENTS IN THE IT INDUSTRY.....	28
5.1	Research methodology.....	29
5.2	Research results	32
5.2.1	Theme 1: Individually constructed understandings of business value (standalone)	33
5.2.2	Theme 2: Prioritisation as shared responsibility across several roles (overarching)	34
5.2.3	Theme 3: Prioritisation factors and indicators (overarching).....	35
5.2.4	Theme 4: Prioritisation and business value estimation techniques (overarching)	37
6	DISCUSSION	39

6.1	Interpretation and comparison of key findings	39
6.2	Limitations of the study	46
6.3	Recommendations for future research	47
7	CONCLUSION	48
	LIST OF KEY LITERATURE	49
	REFERENCE LIST	50
	APPENDICES	55

LIST OF TABLES

Table 1:	Summary of demographic and professional background data of participants	31
Table 2:	Summary of themes and sub-themes as a result of thematic analysis	33

LIST OF FIGURES

Figure 1:	The waterfall process model	4
Figure 2:	Increase in value and decrease in risk through time	7
Figure 3:	Steps in thematic analysis	31
Figure 4:	Most commonly used terms within the proposed definitions of business value in the literature review	40
Figure 5:	Most commonly used terms within the proposed definitions of business value in qualitative research.....	41

LIST OF APPENDICES

Appendix 1:	Interview questions.....	1
Appendix 2:	An extract from a coded interview transcript.....	2
Appendix 3:	Excerpt from the process of grouping codes into identified themes as a step in the conducted thematic analysis	2
Appendix 4:	Automated agent configuration for business value estimation and prioritisation – concept testing	3

LIST OF ABBREVIATIONS

AHP - Analytic Hierarchy Process

AI - Artificial Intelligence

BCR – Benefit-Cost Ratio

GAI – Generative Artificial Intelligence

IRR – Internal Rate of Return

IT - Information Technology

NPV – Net Present Value

PBI – Product Backlog Item

PBP – Payback Period

PO - Product Owner

ROI – Return on Investment

UVP - Unadjusted Value Point

VAF - Value Adjusted Factor

1 INTRODUCTION

Throughout the years, the Information Technology (hereinafter: IT) industry has had a great influence on many modern industries, mainly playing a central part in their transformation (Highsmith et al., 2020). Advancements like automation, cloud computing, and even Generative Artificial Intelligence (hereinafter: GAI) in recent years are just part of the disruption of modern industries, reshaping traditional business models and their long-standing practices (Gupta et al., 2025). These advancements have pushed organisations to learn more, adapt, and invest in digital solutions in order to keep up with the competition and serve customers' needs. This shift in strategy for many organisations meant that they needed to invest in a digital solution, like a software application, in order to stay relevant and competitive on the market (O'Toole et al., 2020), which is why software engineering has become one of the key drivers of business value and innovation for many organisations (Highsmith et al., 2020).

In such dynamic times, it is important to understand what organisations perceive as business value and how they determine what brings them the biggest business value when defining their overall strategy. Since this is a broad topic that spreads across many areas within an organisation, my focus will be particularly on software development as one of the key drivers of business value within many modern organisations. And as mentioned throughout the literature, what is being developed and when depends on what the organisation has prioritised, mainly in terms of what brings the biggest business value (Farley, 2021). Software development within an organisation is a structured process with clear inputs and outputs, often following a certain development model to guide the organisation through the overall process (Dooley, 2017). Recent numbers show that incremental and iterative software development models are becoming more relevant due to the dynamic changes in customer needs and industry trends and the increasing need for organisations to react fast to competition and market needs, ultimately becoming more agile (VersionOne, 2023). This is where Agile software development, along with its frameworks and methodologies, becomes dominant in the IT industry for those organisations who are seeking to improve communication, strive for continuous improvement, improve responsiveness to change, and, most importantly, deliver value – fast and often (Szalvay, 2004). To narrow down my research, I have decided to focus on software development using the Scrum framework, which is one of the most popular Agile frameworks nowadays (Hron & Obwegeser, 2022).

The Scrum framework focuses on certain roles, events, artifacts, and principles that organisations should follow when developing software. When it comes to defining business value, the Scrum framework introduces the product owner as a role mostly responsible for defining business value and prioritising the software development process based on what brings the most business value for the organisation (Kelly, 2019). The role of the product owner faces many responsibilities and challenges within the team; however, mainly, this role is responsible for the product backlog, which is a collection of all the elicited software

requirements that later shape the features of the software that is planned to be built (Schwaber & Sutherland, 2020). However, how product owners decide which software requirements have higher priority over the others is not a clear and straightforward process, even though their decisions shape the direction in which the software is built. One research study shows that only 26.5% of the product owners they have interviewed use a structured prioritisation method when refining their product backlog and analysing the value of requirements (van Daalen & van Solingen, 2024). In addition to this, if prioritisation focuses on business value, then understanding how product owners estimate business value is of great importance to an organisation.

The purpose of this thesis is to understand whether and how product owners estimate the business value of software requirements so that they can prioritise their product backlog in an effective manner. This study may motivate product owners in the IT industry to examine the available techniques for business value estimation so that they can make more data-informed prioritisation decisions. Improving the prioritisation process of product owners would potentially result in increasing the quality of the software development process, the quality of the software product as an output, and improving customer satisfaction as an outcome. Consequently, the goals of this study are outlined as follows:

- to understand how product owners perceive business value and provide a definition of the business value of software requirements
- to explore what product owners have taken into account when prioritising their product backlog and provide a list of prioritisation factors and methods
- to identify whether product owners estimate the business value of software requirements at all, and outline proposed techniques and factors of business value estimation

In addition to the main goals and as an exploratory step to complement the main findings, this thesis briefly explores how an AI-powered project management tool can support prioritisation practices. Regarding estimation of business value, it may be perceived as a structured process; however, from my experience working as a product owner in the IT industry and supporting opinions found in literature, estimation of business value as a practice seems vague and improvisational (Schwartz, 2016). This research should, at a minimum, challenge product owners and industry specialists to understand the complexity behind a concept that many experts take for granted. It should try to provoke and structure a process that will base decisions on actual data instead of intuition. It should potentially encourage decentralisation to shift away from having such a crucial decision depend on only one role, which is that of the product owner. Overall, the findings of this thesis aim to expose the vague definition and complexity of business value, which is often perceived as a given, and the priority based on it is not challenged simply because it comes from a business expert in the team. To achieve this, the following research question has been defined: “How do product owners estimate the business value of software requirements in order to effectively prioritise the product backlog”?

This thesis follows a structured approach, focusing on both theoretical and empirical research, aiming to potentially find insights in the literature, in the in-depth interviews with product owners, and as per my personal experience working as a product owner. For the empirical part, I have interviewed eight product owners from different companies, countries, and industries. The interview transcripts from the semi-structured interviews have been analysed using thematic analysis, in order to find patterns in the answers which would lead to stronger conclusions to the proposed research question. In addition, unstructured self-observation and non-participant observation of other product owners were used as a supplementary data source to provide more inputs to the research on the business value estimation and prioritisation practices of product owners.

This thesis is organised in a way that presents a logical progression from the theoretical foundations of the topic to the empirical research and final conclusions. Chapter two explains the nature of Agile software development and the characteristics of one of its frameworks, which is Scrum. Chapter three then focuses on the product owner, as the role that is ultimately responsible for the maximisation of the business value of a software product. Proposed definitions of business value and techniques of business value estimation are elaborated in chapter four, which also presents some of the main literature insights later discussed and compared to the findings from the empirical research. Then, chapter five explains the research methodology used, as well as the results from the empirical research. The last part of this thesis is reserved for discussion and comparison of the results from the theoretical and empirical research, followed by the limitations of this research, as well as recommendations for future research ideas.

2 AGILE SOFTWARE DEVELOPMENT

Agile Software Development is a set of principles, based on various values, which orchestrates the software development process by supporting the team involved in developing the software with an adequate level of decision-making autonomy, emphasising the importance of collaboration and outcomes that eventually bring bigger value to the stakeholders (Campbell, 2020). An increasing number of companies, even ones not focusing on software development, are getting interested in following the Agile methodologies and frameworks as opposed to the traditional ones used extensively in the past years (Žužek et al., 2020). One reason they're doing this is that Agile has constantly been proving itself over the past years to be effective in building successful products that require shorter time to market, are tailored according to the needs of the end users, and provide more predictability and less waste. Some authors claim that Agile helps in building better products than before (Gower, 2013).

Agile is best defined with the Manifesto for Agile Software Development, created and signed in 2001, although it is believed that Agile methodologies have occurred even before, during the 90s (Szalvay, 2004). As written in the Agile Manifesto, Agile is improving software

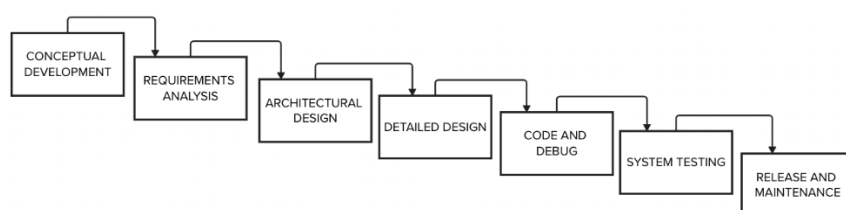
development by focusing on individuals and interactions, creating working software, frequent customer collaboration, and the ability to respond to change (Beck et al., 2001). Agile software development has imposed significant changes to the traditional software development models by introducing Agile methodologies and frameworks that focus on productivity and value (Racheva et al., 2009). The Agile methodologies and frameworks use an iterative mechanism towards software development, producing valuable outcomes with each iteration and allowing software to be developed in increments rather than as a whole (Szalvay, 2004).

Ever since the introduction of Agile, companies have been focusing on Agile transformation in order to produce better results. Experts in the IT industry stand behind the idea of Agile being used to get feedback fast, thus learn fast and improve, whereas learning is one of the main drivers of success for companies in dynamic industries such as the IT industry (Pichler, 2010). Agile advocacy bodies suggest that organisations that do not plan to transform into Agile working entities may face operational challenges in responding to change (Scrum Alliance, 2017). The incremental nature of Agile allows the team to learn more about its customers and the overall market. Consequently, the risk of creating the right product decreases with each increment, as customers learn more about the product, and they are able to provide early feedback, which can steer the product development in the right direction (Szalvay, 2004).

2.1 The transition from traditional to Agile software development

Before Agile, there were the traditional or plan-driven models, also known as the linear sequential models. They are known to follow a strict order in activities, where each phase and its expectations are defined, and each phase should be completed before moving to the next one in the flow (Pressman, 2000). It is a well-structured and designed process, with great focus on planning, analysis, and documentation, as well as thorough verification before anything is released to customers (Dooley, 2017). When speaking about traditional process models for software development, people are mostly referring to the “waterfall” model, where software development is conducted sequentially, following a clearly set order of phases, where the beginning of one phase is dependent on the deliverables of its preceding phase, as shown in Figure 1 (Kuhrmann et al., 2016).

Figure 1: The waterfall process model



Source: adapted from Dooley (2017).

Traditional models, such as the waterfall model, tend to create a mutual understanding and transparency in the requirements, implementation approach, and expectations, thus keeping all parties informed and aligned at all times (Kuhrmann et al., 2016). This is due to the thorough analysis and planning being done at the beginning of the development process, demanding that all requirements are known and analysed before any development happens. This model works well when all requirements are known and are assumed to be stable throughout the whole software development life cycle (Wiegers & Beatty, 2013). Additionally, a great benefit of this model is its predictability, allowing stakeholders to easily predict the project's trajectory. On such projects, estimating the budget and timeline is a stable process with reliable outputs (Kuhrmann et al., 2016).

However, the traditional software development principles are becoming increasingly inadequate for developing modern software, where traditional software development faces 30% project failure rate, compared to Agile software development, which accounts for 10% project failure rate (Mishra & Alzoubi, 2023). This is mainly because of the fast-increasing system complexity, where, due to the advancing technology and the dynamic change in demands, the developed software is expected to be delivered fast and be capable of doing what once had been considered impossible. Additionally, software development models should not be randomly chosen and used but rather tailored to the envisioned software that should be developed (Sommerville, 2015). For example, when developing aircraft software, it is crucial to gather all requirements and specifications prior to implementation (Pressman, 2000). In this case, it is important to note that a more sequential process model is suitable for software development, unlike the development of an e-commerce software, where the specification and software can be developed in parallel. Therefore, when choosing a software development model, it is good to first align on the type of software planned to be developed, the customers and their preferred regulatory requirements, as well as the environment where the software is planned to be used (Pressman, 2000).

In recent years, businesses have faced dynamic changes due to the rapidly changing environment in which they operate (Žužek et al., 2020). In order to keep up with their competition, align with the market demands and opportunities, adapt to the changing economic conditions, and provide the best products and services to their customers, businesses need to act fast and respond to all these changes. And since software has taken part in an increasing number of business operations, software development itself needed to change the pace to provide bigger value to the businesses (Sommerville, 2015). This value puts an emphasis on the customer focus of software development. Mainly, customer involvement, feedback, knowledge, and relationship are important to maintain when developing software if the goal is to develop a high-quality and highly valuable software (Lohan et al., 2011). Having all this in mind, the need for a more flexible and effective software development model emerged, one that provides fast delivery of the software product to customers, thus receiving fast and concise feedback in return that can shape and

repurpose the software development process. This model is known as the iterative and incremental software development model (Szalvay, 2004).

In a software development model that is iterative and incremental, requirements for the complete software are not required to be ready, defined, and structured at the beginning of the software development process, but rather a smaller scope needs to be defined at the beginning of each iteration (Dooley, 2017). An iteration in this sense is a repeating cycle of development that results in producing an increment, or a part, of the software instead of the complete software product. The main goal of an iterative and incremental software development model is to build software in parts and repeated cycles, offering each cycle to end in a set of functionalities to be presented to stakeholders, with the aim of gathering feedback for the completed work and potentially including that feedback in the following iterations (Dooley, 2017). With this model, the first increment of the software is often considered to be the core product, where the essential requirements are taken into account. This core product, if it functions as a stand-alone product, can be tested by real customers who can then evaluate and provide useful feedback that the software development team can consider for the following iterations (Pressman, 2000).

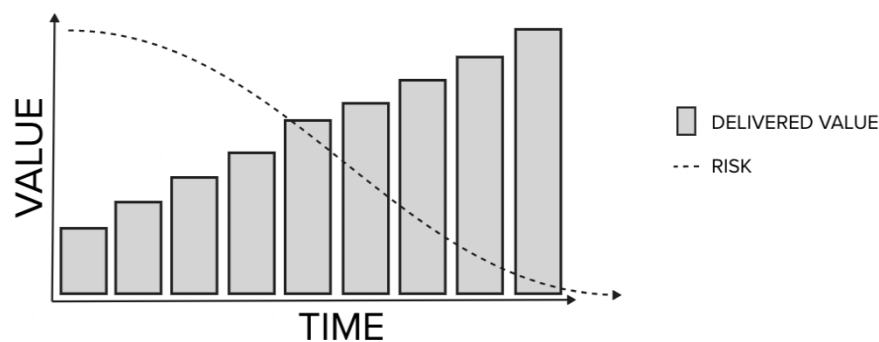
Overall, the incremental and iterative model puts customers in focus by capturing feedback and incorporating that feedback during the software development life cycle. Also, it improves reassurance of the development team that they are on the right track with the product, it lowers impatience with the final result, and helps build understanding and expectations on the software product and its deliverables (Dooley, 2017). And in the centre of this model is the Agile approach, whose main characteristics are iterative delivery of value increments, customer empathy, flexibility, adaptability, and building the right thing (Gower, 2013).

2.2 Delivering value as the main focus of Agile software development

One of the main focuses of Agile software development is value delivery, yet with traditional models, like the waterfall model, software releases happen late in the development phase and therefore delay value (McGreal & Jocham, 2018). One of the main reasons why traditional models are no longer the right fit is because of the impracticality of businesses to come up with solid software requirements that would undoubtedly respond to the business needs after a long period of software development and the release of the software itself (Sommerville, 2015). However, there are occasions on which traditional models are still valid and preferred. Software products that are based on steady and unchanging requirements, where the complexity of the project is low, tied to strict regulatory standards and compliance documentation, often prefer the traditional models. Additionally, it may happen that traditional models are selected due to stakeholders' preference (Kuhrmann et al., 2016).

The rapidly changing environment naturally affects the business requirements to keep changing due to different reasons such as: businesses not being able to predict what will be the software's influence on the working practices, how the software can be integrated with other systems, how the end users will react to the software, and if any value, at all, will be perceived from working with it (Sommerville, 2015). This is where Agile becomes relevant; with its iterative approach in producing and delivering pieces of software, so called increments, and aligning the development based on customer's feedback, Agile practitioners are encouraging the idea for the software to grow through time, therefore allowing changes to be made to the implementation planned based on the changing customers' requirements (Campbell, 2020). When it comes to prioritisation, Agile, in contrast to plan-driven software development, focuses on prioritising the features with the highest business value first (Racheva et al., 2009). Overall, Agile methodologies contribute significantly to creating business value as the software development process is organised in a way that is mainly focused on creating the right product and increasing customer collaboration and overall satisfaction (Racheva et al., 2010a).

Figure 2: Increase in value and decrease in risk through time



Source: adapted from Campbell (2020).

The incremental nature of Agile allows the team to learn more about its customers and the overall market. Consequently, the risk of creating the right product decreases with each increment, as customers learn more about the product, and they are able to provide early feedback, which can steer the product development in the right direction (Gower, 2013). Figure 2 shows the effect of Agile software development, where value increases and risk decreases over time (Campbell, 2020). And once teams decide to leverage the advantages of Agile, they might also choose a specific Agile methodology or framework to follow for a more structured software development process.

2.3 Agile methodologies and frameworks

Upon setting the general principles for Agile software development, experts have come up with various methodologies and frameworks that can be utilised to focus on value delivery

instead of project completion. Some of the most well-known ones are: Scrum, Lean, “Extreme Programming or XP”, Crystal, DSDM, Feature Driven Development, and Kanban (Campbell, 2020, p. 9). All of these methodologies and frameworks organise the work in short development cycles, often referred to as iterations or sprints, and focus on enriching the product with more features each iteration (Dooley, 2017).

The Agile methodologies and frameworks should be chosen based on the project specifications and the team’s preferences. However, there is no strict rule that only one particular methodology should be followed. In fact, Agile practitioners are encouraged to combine and make use of several Agile methodologies and frameworks for a project as long as it is a controlled process and provides them with the right benefits (Campbell, 2020). For example, Extreme Programming (XP) is an Agile methodology that is complementary to Scrum and can be used in combination with the goal to improve the technical practices within the software development team and focus on continuous process improvement (Rubin, 2012). Kanban is also used in combination with other methodologies, offering a visual representation of the project tasks and progress, enabling better project management. On the other hand, Crystal methodology focuses more on the people and the team communication, guiding the team in improving their productivity while focusing on what is working and helping them understand what is not working so well (Campbell, 2020).

When it comes to customer-centricity, value-based development, and team empowerment, Lean and Scrum frameworks are known to be the most popular choices for developing modern software products. Out of all these methodologies and frameworks, the Scrum framework distinguishes itself as the most practised Agile approach of all (Hron & Obwegeser, 2022). For the purposes of this master’s thesis, I have chosen Scrum as a framework of interest, due to its popularity, strong focus on business value, and my experience with this framework on the numerous projects I have worked on.

2.4 The Scrum framework

As an Agile framework, Scrum is focused on delivering software to customers with less cost, in less time, with higher quality, and aligned to the customer’s wishes and feedback. As per definition: “Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems” (Schwaber & Sutherland, 2020, p. 3). Scrum highlights the importance of developing software in smaller teams to improve efficiency and encourage continuous improvements inspired by frequent reviews and feedback from the client (Hron & Obwegeser, 2022). Having this in mind, Scrum revolves mainly around the customer, focusing on continuously delivering customer value instead of just producing output based on previously defined requirements (Larman & Vodde, 2016).

The Scrum framework is based on empiricism, and as such, it values transparency, adaptation, and inspection (Schwaber & Sutherland, 2020). While transparency is important

to keep everyone in the development loop, inspection and adaptation emphasise the collaboration between the team developing the software and the stakeholders, where, during meetings, such as the sprint review, the development effort and the overall scope are being reviewed and adapted based on the feedback received. This collaboration throughout the whole development process is what improves the overall development process, since the team can review the value they have delivered to the stakeholder until a certain point in time and ideate on how to additionally maximise the value yet to be delivered (Schwaber & Sutherland, 2020).

The Scrum framework works best when the team working on building the software is a cross-functional team that is responsible for designing, developing, testing, and maintaining the software while working in timeboxed periods called sprints. The roles that comprise the Scrum team are the product owner, the Scrum master, and the development team (Rubin, 2012). These roles, as guided by the Scrum framework, organise their work with respect to the main Scrum events, which are the sprint, sprint planning, daily Scrum, sprint review, and sprint retrospective. Furthermore, the framework is grounded on three main artifacts, which are the product backlog, sprint backlog, and increment (McGreal & Jocham, 2018). Generally speaking, the product backlog describes what needs to be built, usually created once a product discovery process has resulted in identifying product requirements worth building (Cagan, 2017). It comprises all the requirements and configurations of the envisioned product, ordered in a matter of priority and dependency, and it represents the single source of truth for the whole team (McGreal & Jocham, 2018). The sprint backlog is a smaller subset of items already part of the product backlog. It is a list of tasks the development team has been well introduced to and is confident that they can finish within a sprint. The items entering a sprint backlog are thoroughly analysed during sprint refinement and planned and prioritised during sprint planning, guiding the development team towards the defined sprint goal (Rubin, 2012). The product and sprint backlog are on the input side of software development. However, the increment is a Scrum artifact that represents the output or the product deliverable, which comes as a result of the tasks completed within a sprint and all preceding completed tasks. The increment should be functional and usable once completed, although it is a strategic decision whether that increment would be released to customers directly (SCRUMstudy™, 2016). Eventually, each increment should contribute to the overall value of the product.

Nowadays, IT practitioners believe that if technology companies want to be successful, they need to put the focus on consistent product innovation, which is mainly done by constantly creating value for their business, their stakeholders, and the final users (Cagan, 2017). The Scrum team is a self-organising team, yet the product owner is the one mainly focused on the product vision and goals, working on prioritising the product backlog and driving all conversations with the stakeholders and potentially the end users. Therefore, the product owner is eventually responsible for the delivery of value (Pichler, 2010). Consequently, the following chapter will look into the definition of the product owner, their responsibilities

and challenges, so as to understand their perspective on business value estimation of software requirements.

3 THE ROLE OF PRODUCT OWNER IN SOFTWARE DEVELOPMENT

Unlike in traditional software development, where the end user of the product is represented by the customer, in software development in an Agile environment, the end user can be represented by an actual customer, a product owner, a product manager, or any customer proxy that is working towards increasing the benefits of the end users (Lohan et al., 2011). Within Scrum, this is usually the product owner, and it is this person's responsibility to make sure that all team members and stakeholders are aligned on the product vision and pursuit of common goals (Cohn, 2006). When it comes to building the product, the product owner is the one deciding what the team should work on next within a sprint backlog, what's part of the overall product backlog, and how the items within are ordered (Schwaber & Sutherland, 2020). Regardless of the prioritisation method used, the product owner usually aims to organise the work so that functionalities with the highest value are tackled first. Such decisions are expected to provide a solid return on investment for the company (Cohn, 2006).

Planning of release or preparing for launch, including and managing users, customers, and other stakeholders, managing the product life cycle, managing the budget, participating in Scrum meetings, and collaborating with the Scrum team are all part of the responsibilities of a product owner (Pichler, 2010). Depending on the company, the project, and its setup, different product owners might have different responsibilities. However, what is generally perceived as challenging is when a product owner is taking ownership of someone else's idea, like a business owner, and having to represent it and help it succeed. In such cases, a good product owner not only takes ownership of the idea, but they also help the whole Scrum team take ownership of the idea and build the product as if it were their own (Patton & Economy, 2014).

The collaboration between the product owner and the Scrum team happens on a regular basis. The product owner is even responsible for monitoring the progress made towards the set goals, preferably during the ongoing sprint, which is discussed during the sprint review. Having this in mind, the product owner follows the ongoing work during the current sprint, while analysing the past sprints and using all gained knowledge to optimally plan the upcoming sprints (Schwaber & Sutherland, 2020). Apart from the development team and the Scrum master, the product owner should work hard to maintain relationships with end users, customers, and executives. This is important as nurturing those relationships improves the chances of the product owner becoming a more knowledgeable and trustworthy domain expert (McGreal & Jocham, 2018). Aside from these roles and depending on the project, the product owner may need to cooperate with other roles as well, such as technical architects, product managers, information security officers, and so on (SCRUMstudy™, 2016). The

product manager and product owner have overlapping responsibilities when it comes to strategic activities, such as release planning, creating a roadmap, industry and domain analysis, customer research, product discovery, and others. This raises the question of the purpose and the need for both roles to be working on the same project (McGreal & Jocham, 2018).

3.1 Distinction between the product owner and product manager

As both product owner and product manager are engaging in strategic activities, and maximising value requires strategic decision making, it is important for me to make the distinction between the two roles and clarify why I have chosen to analyse the product owner's perspective on business value estimation. There are different views on whether the product owner is just an Agile product manager. Some argue in this direction, others say that product ownership is much more than product management, while some state that product ownership is just a subset of the product management role (Rubin, 2012). Cagan (2017) goes a step further, saying that this similarity depends on the type of company. For example, he highlights the importance of the product manager to be the product owner when a company is considered a product company. Splitting these roles would indicate some difficulties along the way, like disabling the Scrum team from innovating and working towards the business goal. Nevertheless, he agrees that the product owner's role is just a subset of the responsibilities of the product manager (Cagan, 2017).

Overall, the product owner is accountable for the success of the product, which is also applicable to the product manager as a role (Patton & Economy, 2014). However, depending on the working model used by the company, including a product owner and product manager in the same team usually means that both roles will focus on the product, but on different levels. The product manager will focus on continuous product discovery, which is understanding the general needs and wishes of the customers and identifying potential solutions, while the product owner will transfer the solution proposal into the product backlog and communicate it with the development team (Torres, 2021). As part of my thesis, I am mainly focused on Scrum as a preferred Agile framework. Given that perspective, some authors state that product management is way bigger than the focus of Scrum. And since the product owner is mainly a Scrum-introduced role, the product owner only tackles part of the product manager's area of interest (McGreal & Jocham, 2018). Despite their overlapping characteristics, my focus is on the product owner and their perspective on business value estimation; therefore, I will analyse the product owner as a role, its responsibilities, and, more interestingly, its challenges.

3.2 Main challenges of the product owner

When developing a product, it is highly important to have the team be aligned on the product vision, which is one of the responsibilities of the team's product owner. And usually, the

bigger the team, the more challenging it is for the product owner to establish and maintain this alignment (Cohn, 2006). One of the main challenges of a product owner is to effectively practice the role of a communicator, which is one of the essential skills required from a product leader. Even though a good product owner is considered to be a good communicator, communication between the team or with stakeholders outside the team can be quite a challenge for a product owner, especially if the product owner is not aware of the business priorities or does not understand the technical perspective of the product (Magnussen et al., 2024). Furthermore, the COVID-19 pandemic imposed additional obstacles in the work dynamics of the product owners (Mendonça dos Santos et al., 2023). The remote way of work enforced by the pandemic had heavily affected and limited the communication within the team, especially the spontaneous communication that happens naturally in an office, from which product owners benefit in terms of consultation with the development team, quick review, and unofficial brainstorming (Mendonça dos Santos et al., 2023). In Agile software development, where individuals and interactions, as well as customer collaboration, are preferred as opposed to documentation, processes, and tools (Beck et al., 2001), it is evident that the pandemic challenged product owners working in an Agile environment, requiring them to show adaptation and resilience (Mendonça dos Santos et al., 2023).

Another interesting challenge that product owners face, to a different degree depending on the project and company setup, is the ownership and decision-making autonomy. On some projects, a product owner may be assigned to a product's idea that is not originally theirs, so the challenge there is to take ownership of the idea and make strong efforts to make that product successful. In such a scenario, it is highly important that product owners identify and surround themselves with the right decision makers in order to help them make good product-related decisions (Patton & Economy, 2014). However, decision-making in software development is not straightforward, as product owners need to have in mind the business objectives, the technical perspective, and the customer value when deciding and guiding. Unfortunately, some product owners struggle with the power of decision-making, especially in companies that have a hierarchical structure with many management layers, where product decisions come from higher management positions (Niva, 2021).

Having a great overview of the product and planning the work for the team is a challenge on its own. Product owners need to be thorough in their research and careful when specifying the requirements that the development team needs to implement (Magnussen et al., 2024). This means that they constantly need to learn about the domain, the customers, as well as the capabilities of the development team, in order to plan the work accordingly (Niva, 2021). However, the need for continuous learning does not revolve only around the ongoing product work, but is also triggered by the dynamic changes of the software development industry. One such major event that shook the software development industry is the emergence of GAI and various AI tools that automate and optimise tasks and workflows (Gupta et al., 2025). Although some early evidence has shown improved processes of requirements gathering and user requirements analysis using GAI, the bright potential of GAI requires

product owners to learn more about this technology and experiment with GAI tools as part of their responsibilities and tasks (Gupta et al., 2025).

The product owner must clearly communicate the product vision and strategy, usually by creating a product roadmap that clearly communicates the product features and priorities to the target audience (Lukassen & Schuurman, 2023). A roadmap is an overview, a planning artifact of the high-level features that the product should consist of, representing the evolution of the product development and its releases as milestones to be achieved through time. The created roadmap should provide guidance and encourage discussions within the scrum team and the stakeholders (Pichler, 2010); however, once created, the stakeholders usually view it as a commitment. And it's rarely the case that a roadmap, as envisioned and created initially, can be delivered in time and without any changes in the scope or the project resources (Cagan, 2017). Product owners need to be able to adapt to changing requirements if they want to keep up with the changing market and customer needs. This requires well-established prioritisation skills, where the product backlog is reviewed and reprioritised based on factors that the product owner finds important (Lukassen & Schuurman, 2023). Eventually, this prioritisation is not a one-time thing as the product owner needs to make sure that the team is always working on the highest-valued functionalities (Cohn, 2006). In order to do the requirements prioritisation right, product owners are often advised to prioritise based on business value. However, this advice is considered confusing and vague since authors and experts are not entirely aligned on the definition of business value and what it actually represents (Racheva et al., 2009), only that someone from the business, like the product owner in Scrum, should determine what is valuable and prioritise accordingly (Schwartz, 2016). In summary, the product owner is responsible for collecting and prioritising software requirements from stakeholders in order to define the product according to the stakeholders' needs and expectations. It is the product owner who is ultimately responsible for managing a product's requirements, and their ability to gather, understand, and prioritise the requirements can highly affect the overall success of the product (Project Management Institute, 2021).

3.3 Understanding software requirements

Product owners are expected to clearly process the elicited software requirements, which is why it is important to understand what software requirements represent and how they are managed. Unfortunately, some experts have struggled to agree on only one definition of software requirements, resulting in many of them bringing a new aspect to the definition, which is an additional motivator to find a common ground for understanding what software requirements represent (Wiegiers & Beatty, 2013). Generally speaking, in software development, software requirements answer the question of what should be built (Kelly, 2019). A requirement expresses a need, which can be, although it is not always, constructed from the customer's perspective. Having this in mind, a software requirement can be a business requirement, a stakeholder requirement, a solution requirement (functional and

non-functional), and a transition requirement (International Institute of Business Analysis, 2015). Other authors propose a different classification of requirements, such as dividing them into functional and non-functional requirements, or primary and derived requirements, or business and technical requirements, and so on (Aurum & Wohlin, 2005).

Requirements can also be seen as the outcomes of stakeholders' decisions on what should be the functionality and quality of the software being developed (Regnell et al., 2001). They can be defined based on the level of detail and the perspective from which they are described. Development teams, when using a traditional model like the waterfall model, think that a requirement can be either a user or system requirement, the former explaining the services provided to the user, whereas the latter would be more about what the system itself can and should do (Sommerville, 2015). This two-way perspective can also be spotted in IEEE's glossary, where two definitions for requirements are provided, the first being from the user's perspective, whereas the second is from the system's perspective. Both definitions explain the requirement as a condition or capability, aiming to satisfy different objectives either from the user's or system's perspective (Institute of Electrical and Electronics Engineers, 1990). Some authors capture the idea behind software requirements by defining them as "...specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system" (Sommerville & Sawyer, 1997, p. 4). Although this may seem like a suitable definition, authors emphasise that requirements should not describe how the system should behave, but rather what it has to do in order to satisfy users and other stakeholders' needs (Aurum & Wohlin, 2005). In this direction is the proposed definition by the Project Management Institute (2021, p. 48), which explains that "A requirement is a condition or capability that is necessary to be present in a product, service, or result to satisfy a need". Since there are different perspectives on the definition and meaning of software requirements, for the purpose of my thesis, I will be relying on the one from Sommerville and Sawyer.

3.3.1 Software requirements elicitation

Software requirements elicitation is one of the main activities of product owners. Generally, it is described as "...the process of seeking, uncovering, acquiring, and elaborating requirements for computer based systems" (Zowghi & Coulin, 2005, p. 19). Software engineering experts sometimes use the term requirements gathering, yet some authors emphasise a clear distinction between the terms gathering and eliciting, explaining that requirements gathering assumes that the customers or stakeholders already have a clear set of requirements ready to be collected and analysed by the product owner or business analyst (Wiegers, 2021). In that regard, requirements elicitation is more than simply gathering requirements as it covers many activities connected to communication and learning more about the domain, continuous discovery of the needed functionalities, active observation, and understanding of customers' needs (Aurum & Wohlin, 2005).

During requirements elicitation, the focus should be on the user of the software, what they need from the software, how they can use it to address their pain points, what feedback they have for the current processes, and what they would like to improve, etc. (Lohan et al., 2011). Most times, the requirements come from stakeholders other than the users themselves, which is why it is important for the requirements to be based on thorough user research conducted through interviews, reviews, conversations, prototype evaluations, and iterative feedback on small increments of working software (Wiegers & Beatty, 2013). Traditional models propose that software requirements should be defined in the early stages of the software development lifecycle, before any development activity begins (Kelly, 2019), whereas requirements elicitation happens throughout the whole development life cycle, since requirements are likely to keep changing through time (Rigby et al., 2018). Changing requirements and business goals are expected by a product owner working in an Agile environment, which is why a good product owner should be skilled in prioritisation and reprioritisation.

3.3.2 Software requirements prioritisation

Software requirements prioritisation is performed on a group of requirements, or the complete product backlog, once software requirements have been elicited, and it is an important mechanism within Agile development, significant for maximising the business value of the product while managing the changing requirements within the product backlog (Bakalova et al., 2011). It is a decision-making practice that helps the team focus on the right thing to build, at the right time, thus reducing development cost. Moreover, it helps in managing the subjective views of the stakeholders, requiring them to agree on the priorities and align on the delivery expectations (Gambo et al., 2018). Software requirements prioritisation is one of the key factors for the success of a software product, significant to be conducted correctly, at the right time (Asghar et al., 2017).

The product owner is responsible for prioritising the product backlog, where all requirements are listed. Even though customers want to have all requirements implemented, often due to limited resources such as time and cost, the implementation of all requirements is not possible within the desired timeframe. Therefore, prioritisation helps in setting the development focus on the right features first, thus avoiding conflicts, setting the right expectations with the stakeholders, planning releases that deliver value, and helping the product owner make certain trade-off decisions (Wiegers, 1999). Some authors say that this ranking and shifting of requirements in the list back and forth according to their priority does not have to be a formal activity and can rely on the product owner's intuition and decision (Cohn, 2006). Even though this may seem to work for some projects, it gets harder to make the best priority decisions once more stakeholders become part of the project. This is because, as per human nature, each stakeholder will have different goals and requests; therefore, each stakeholder will require a different prioritised list of requirements (Wiegers, 1999). Furthermore, organisations might have a portfolio of work larger than they can

handle; therefore, the need for a prioritisation method is quite emphasised in such cases (Harris, 2017).

Initial and continuous prioritisation is executed within each software development project; however, the prioritisation method may differ, as well as the factors that influence the prioritisation decision. Cost, time, workforce, dependencies, and clearly defined goals - these are just part of the factors product owners base their prioritisation decisions on (Al-Ta'ani & Razali, 2013). This diversity in prioritisation methods and factors of prioritisation motivates me to research the methods and factors that the product owners, interviewed as part of the empirical research, base their prioritisation on. As for prioritisation methods found in literature, here are some known to be widely used by product owners (Khan et al., 2015):

- Numerical assignment
- MoSCoW (Must Have, Should Have, Could Have, Won't Have this time)
- Simple Ranking
- Bubble Sort
- Binary Search Tree
- Hundred Dollar Method
- Analytic Hierarchy Process (hereinafter: AHP)
- Minimal Spanning Tree

It should be noted that the above list comprises prioritisation methods, techniques, and mechanisms. However, due to consistency, they will all be regarded and used under the umbrella term of prioritisation methods. That being said, some of these prioritisation methods have been thoroughly analysed, concluding that not all of them are suitable for an Agile project. Also, many of these methods do not take into account some of the most important factors to be considered when prioritising, which are time and human behaviour (Rida et al., 2017). Additionally, most prioritisation methods proposed are not adequate to be used for backlogs with a large number of requirements (Kukreja et al., 2013). Prioritisation methods like Pairwise analysis, the MoSCoW method, or the hundred-dollar method rely on the intuition and assumptions of the people involved in the prioritisation process, whereas the overall goal is to avoid subjectivity and provide a more explicit and objective decision-making process (Racheva et al., 2008). Some authors, after experimenting with various prioritisation methods, agree that, from most methods mentioned above, AHP is the most suitable one, as it is most fault-free compared to all others, it encourages consistency checks, and provides fairly reliable results (Khan et al., 2015). However, there are also authors arguing that this method takes more time than the others, it is the hardest to use, and is not appropriate for a bigger backlog with requirements (Khan et al., 2015). Some authors criticise the existing prioritisation methods due to their inability to perform in an effective, efficient, and scalable manner (Gambo et al., 2018). Therefore, they have tried to modify or combine the existing methods in order to improve the prioritisation of requirements, especially when working with a big backlog consisting of a large number of

requirements and many stakeholders who influence the decisions about the software being developed. Other than the methods mentioned above, there are many other methods, models, matrices, and techniques that can be used to help the decision-making process when prioritising the product backlog. Some worth mentioning are: impact versus effort matrix, Kano model, Weighted Shortest Job First, Story mapping, Rice model, risk versus return matrix, Weighted scoring, Opportunity scoring, etc. (Trieflinger et al., 2021).

Racheva et al. (2010) analyse the methods for reprioritisation of a product backlog, based on the main factors like risk, business value, effort estimation, external change, learning experience, and project constraints. In their proposed model, where the client/customer is an important decision maker, business value is one of the most important factors the reprioritisation is based on. This indicates that, in order to use prioritisation methods (such as the MoSCoW method), the product owner is expected to already have knowledge about the value each requirement provides so that they can use this knowledge in using the proposed prioritisation methods and eventually prioritise the requirements based on their business value (Kukreja et al., 2013). Consequently, the next section will address the business value estimation of software requirements, which is a practice that should help product owners understand the value of software requirements.

4 BUSINESS VALUE ESTIMATION OF SOFTWARE REQUIREMENTS

As previously mentioned, the software requirements in a product backlog are prioritised based on several factors, one of which is business value. Since the product owner is accountable for the product backlog, this person should have information about the business value of the requirements that have transitioned into product backlog items (hereinafter: PBI). Furthermore, they need to acknowledge the amount of business value each software requirement brings to the overall product, which can be estimated. Generally, estimation is defined as the calculation or judgement of something, consequently providing a certain or approximate result based on the size, weight, time, value, or amount of whatever is being analysed and estimated (Dinwiddie, 2020). For projects in particular, estimation provides an approximation of effort, duration, and cost based on gained knowledge, experience, etc. And by following the concept of the cone of uncertainty, the more time passes, and the more information and experience is gained by the team working on the project, the cone shrinks, and therefore the uncertainty decreases (Harris, 2017). That means the estimations become more accurate and reliable than before.

The need for value estimation of software requirements becomes more important with the rise of the customer-centric focus in software development, which is specific to the Agile methodologies (Lohan et al., 2011). In order to develop a high-value product for the customers, a value-based approach should be undertaken in the planning activities for the development of that product. Part of this approach is to quantify the value of a product so

that the benefit of building that product can be justified against the resources invested (Mohamed & Wahba, 2008). When an organisation is planning the development of a software, they need to consider the needs, desires, and pain points of their users; therefore, simply gathering requirements and scheduling their implementation is not enough. Product owners need to find the best combination of the product scope, the time or schedule in which it will be realised, as well as the cost associated with the realisation of the scope and its effect on the overall value gained from that scope (Cohn, 2006). Consequently, business value should not only be analysed quantitatively, but also through its qualitative factors. Therefore, it often happens that business value estimation is based on human judgment due to the knowledge gained regarding the customer's domain and preferences (van Daalen & van Solingen, 2024). However, this judgement should be transparently communicated and presented from the product owner to the development team and the stakeholders, so that they are aware of how this decision on the business value of requirements has been formed, and with this create a common ground of understanding and therefore strive towards mutually defined goals (Racheva et al., 2010b). And in order to do this right, product owners need to base their decisions on solid data and well-thought-out estimates. To find out how to best estimate the business value of software requirements, I first need to understand what business value really is. Unfortunately, many experts in the IT industry are not entirely aligned on what business value actually represents (Cohn, 2006). Therefore, it is important to analyse how business value is defined in literature and, later on, in the practical experience of the product owners who have taken part in the empirical research of this thesis.

As part of the literature review, aside from the quest to find ways in which different authors defined business value estimation and its importance, I will go about explaining some of the existing techniques to estimate the business value of software requirements in order to better prioritise the product backlog. But before going into distinct techniques described in literature, it's interesting to acknowledge the different aspects that authors have taken regarding the decision-making process that leads to prioritisation of software requirements. For example, some authors go about describing the initial ranking of a group of software requirements that helps in their prioritisation, stating that this process can and usually does happen in the head of the product owner, relying completely on the intuition and knowledge of that person (Cohn, 2006, p. 54). This intuition-based prioritisation is also analysed in one research study, which shows that 47% of the product owners they have interviewed base the prioritisation on individual preferences and intuition (van Daalen & van Solingen, 2024). Other authors concluded that Scrum teams take part in the prioritisation process with inputs based on experience and implicit knowledge, usually making guesses about the customer's needs and how each requirement would be beneficial to the customers (Lehtola et al., 2004). On the other hand, Harris (2017) criticises organisations that go about delegating estimations to an "expert" in a certain field, leaving that person to come up with adequate estimates based on their experience and knowledge. However, due to cognitive bias, which may affect the "expert" in being overly optimistic, basing the estimates according to their capabilities and knowledge or even comparing the current estimates to ones from a previous project

rather than its actual results, can easily result in an estimation of poor quality (Harris, 2017). Bakalova et al. (2011) conducted research on the prioritisation practices used in practice and in literature, and from the interviews with information IT experts, they found out that, in practice, business value is one of the dominant criteria for requirements prioritisation. Furthermore, their findings emphasise the occurrence of negative value and risk, two important factors that affect the prioritisation process and are connected to the negative value or risk of not implementing a particular feature. However, when it comes to business value estimation, the findings of these authors state that business value is estimated by the customer (Bakalova et al., 2011, p. 186).

As per Scrum, the product owner is the person responsible for maximising the product value (Schwaber & Sutherland, 2020). Customers gain value from the product when their customer needs are fulfilled; therefore, it is important to plan the development team's work well so that the high-value PBIs are implemented and released first (Pichler, 2010). This plan is a continuous process of Product Backlog prioritisation, which is a responsibility of the Product Owner. Having this in mind, the product owner should be able to estimate the business value of requirements, along with other aspects like risk, cost/size, and dependency, to decide on each requirement's importance, to eventually prioritise the product backlog effectively (McGreal & Jocham, 2018). This next part will explore the definition of business value, especially in terms of software development.

4.1 Definition of business value

In management and financial economics, the term business value is informally known as a combination of all forms of value that influence and define the success, sustainability, and well-being of a company (Racheva et al., 2009). Whereas the Project Management Institute (2019) analyses different interpretations of the term business value, defining it in a project-related context, where the business value is the benefit to the stakeholders, provided as a result of the processes and outcomes from the project team. This benefit is seen through the prism of tangible elements like monetary assets, stockholder equity, utility, market share, etc., and intangible elements like goodwill, brand recognition, public benefit, strategic alignment, and reputation (Project Management Institute, 2019). However, every project is different; therefore, the identification of tangible and intangible benefits really depends on the project itself, its stakeholders, and its strategic objectives.

When it comes to Software Engineering, business value revolves around the customer, so when organisations focus on delivering value to the customers, they eventually create value for themselves and their stakeholders (Thompson, 2014). Companies realise they have succeeded in creating business value when they receive a proportionally bigger return on investment as a result of their initial investment in offering a certain product or service (Schlenker & Matcham, 2005). Nevertheless, it seems that the definition for business value is perceived as vague, and many IT practitioners base the definition of business value on the

context of the project, the client, and other factors (Racheva et al., 2010a). On the other hand, some authors state that they cannot expect one general definition of business value, exactly because it varies across different types of projects and organisations, affected by the project settings, the needs of the customers, and the market position (Bakalova et al., 2011). It is thought to be a broad concept, and its understanding largely depends on the context and perspective (van Daalen & van Solingen, 2024). Having the customer as one of the main stakeholders, it seems important to define the business value as seen from their perspective. In this direction, there's an elaborate definition by Harris (2017, p. 11), stating that: "The business value of software is proportional to the customers' willingness and desire to use the software in such a way that the organization can benefit from that use by generating revenue, reducing cost, or meeting some other goal more effectively." Then, there are authors who look at business value only from the financial perspective. Having this in mind, Sward (2006, p. 6) provides the following definition: "Business value is the benefit for business units and the enterprise as a whole, represented in dollar terms, that is a result of IT solution or services, as evidenced by one or more of the following:

- Direct contribution to the corporation's market position or revenue
- Deliverables and results that support solving customer business needs and challenges
- Customer cost savings or financial benefits
- Examples of technology investment that advance the industry."

Some authors are criticising the ambiguity and vagueness of the definition of business value, emphasising how Agile practitioners are determined to prioritise by business value and how the general understanding in Scrum is that someone from the business sector should be responsible for quantifying the business value of a feature. One such author is Schwartz (2016, p. 90), who proposes the following definition: "Business value is a hypothesis held by the organization's leadership as to what will best accomplish the organisation's ultimate goals or desired outcomes". He goes on describing the leadership's hypothesis as not a single plan of activities, but a set of business value indicators that eventually add up to the desired goal. Outcomes like revenue generation, customer satisfaction, risk management, brand-building, and others are perceived as a source of business value that the team seeks to deliver and maximise (Schwartz, 2016). However, based on the Project Management Institute (2021, p. 236), business value is defined as "the net quantifiable benefit derived from a business endeavor that may be tangible, intangible, or both." Further on, they emphasise the different types of stakeholders an organisation or project can create value for, as well as the different perceptions of value, seen as from the perspective of different stakeholders like the organisation itself, the customers, the society etc. In this direction, the Project Management Institute (2021, p. 7-8) shares the following: "Examples of ways that projects produce value include, but are not limited to:

- Creating a new product, service, or result that meets the needs of customers or end users;
- Creating positive social or environmental contributions;

- Improving efficiency, productivity, effectiveness, or responsiveness;
- Enabling the changes needed to facilitate organizational transition to its desired future state; and
- Sustaining benefits enabled by previous programs, projects, or business operations.”

All of these definitions focus on specific areas; however, most of them emphasise the benefit that a business or organisation gains from completing a project. As per McGreal and Jocham (2018, p.135), business value can be gained from the release of a single feature, as they define business value as: “The value created from implementing the feature: revenue, cost saving, customer retention, prospect customers, and future opportunity”. Another common theme is the customer, as one of the main stakeholders of an organisation, and their willingness to use a certain software product that is tailored to their needs and desires. Certainly, the organisation itself and the customers are not the only stakeholders, which is why some definitions also mention the whole industry (Sward, 2006), where stakeholders like competitors and suppliers are taken into consideration, as well as end users and the general public (Project Management Institute, 2021). When speaking about the benefits of the organisation, the financial aspect is undoubtedly the main focus. Goals like revenue generation (Schwartz, 2016), reducing cost (Harris, 2017), the organisation's market position, the customer cost savings or financial benefits (Sward, 2006), and, overall, any net quantifiable benefit (Project Management Institute, 2021) are at the centre of the business value definitions. The effectiveness of meeting those goals is also emphasised by Harris (2017) and the Project Management Institute (2021).

While the Project Management Institute (2021) highlights the importance of business value accounting for both tangible and intangible benefits, Schwartz (2016) keeps a more open definition focusing on the organisation's ultimate goals or desired outcomes, structured as a hypothesis by the organisation itself. This points out that what constitutes business value differs from one organisation to another, so each organisation defines its own set of tangible and intangible benefits. The need to focus on outcomes instead of outputs when talking about business value is discussed by other authors as well. Rather than the organisation thinking about the outputs they deliver throughout the product life cycle, which can be represented by the software features they deliver, they should focus on the actual outcomes they create. The outcomes are defining the impact they have on the lives of their customers and how their perceived value reflects on the business, its profitability, and sustainability (Torres, 2021).

The previous definitions mention the business benefits and the customer benefits as separate streams of value, which brings the question of how business and customer value correlate. Furthermore, the definition of the Project Management Institute (2021) mentions the end users along with the customers, also clarifying how the end user is not necessarily a customer, as the end user is the one who will end up using the actual product, while the customer is the one who funds or requests that product. Such an example is the purchase of an enterprise software, where the company management team decides to buy a software product, whereas operational company staff are meant to be working with that software. For

the sake of simplicity, we will use the customer and end user synonymously for the remainder of this thesis. Generally speaking, customer value is defined by what the customer perceives as valuable (Project Management Institute, 2021). Authors categorise the business value or benefits as internal value, whereas the customer value is perceived as external value. So, in essence, an organisation can set certain goals that will directly increase customer value; however, goals like revenue, profit, and market share are desires that customers see no value in, and therefore, they are internal business benefits to the organisation itself (Highsmith et al., 2020). Further emphasis is put on the contradiction between the business and customer value, in the sense that there might be a software feature or requirement that the customer desires, but the business does not find value in providing it. In fact, there might be times when the decision to increase business value comes with a consequence of decreasing customer value. For example, an organisation may want to increase business value by cutting down costs on servicing the least profitable customers, which would in turn decrease the customer value of those customers (Schwartz, 2016). Finally, Sward (2006) captures the balance between all types of value by clarifying that business value is defined best when the organisation takes into consideration what is perceived as valuable to the end users, to the customers, but also to the organisation itself.

All in all, companies need to refer to their business model in order to create sustainable business value (Schlenker & Matcham, 2005). The business model of a company defines how the company combines the customer segment, value proposition, channels, customer relationships, revenue streams, key resources, key activities, key partnerships, and the cost structure to create and capture value (Osterwalder & Pigneur, 2010). Much of the company's business model revolves around the customer, and since a customer's perception of value can change over time, companies need to keep adjusting their business strategies towards creating bigger business value (Schlenker & Matcham, 2005).

4.2 The purpose and techniques of business value estimation

Earlier in this chapter, it was mentioned that business value estimation is important for product owners as it helps them plan the work better and optimally prioritise the product and sprint backlog. Aside from planning purposes, organisations need to measure the business value they have provided to understand the impact of their software on the business and all stakeholders. "Creating software is about delivering business value. Without some measure of business value, it's hard to determine whether the software has any" (Pugh, 2011, p. 279). Essentially, estimation is a valuable tool that product owners can use to predict and make more confident decisions that will continuously help them check on the progress and whether they are achieving their goals (Dinwiddie, 2020). This section will focus on the overall purpose of business value estimation, as well as expert opinions on whether business value should be estimated at all.

When it comes to decision-making about requirements prioritisation based on business value, some authors have identified the need for a systematic approach that would lower the subjectivity and ensure an explicit and objective decision-making process (Racheva et al., 2008). On the contrary, some authors show scepticism when it comes to measuring and calculating business value to make valuable decisions for a product. One such author suggests that if product owners try to quantify the value by calculating it using some mathematical formula, then they will focus on the numbers and will base their decision on an absolute judgement, strongly believing that they have made the right decision. Instead, estimation techniques should help product owners get to relative judgement, which will provoke different perspectives and discussions that would ideally lead to better decision-making (Torres, 2021). Moreover, authors suggest that product owners should find the correlation between the inputs and outcomes and define the measure of success, which will guide the decision-making prioritisation and will be a good ongoing indicator of the overall development progress. Again, the measures of success do not completely quantify the internal and external benefits of an organisation but rather set a threshold for relative comparison while prioritising (Highsmith et al., 2020). Then there are authors who believe that certain areas of business value of each requirement can be estimated, but it will be too time-consuming to perform it on all requirements, so the results will delay the process of reprioritisation. Hence, the development team will be blocked by waiting for the product owner to decide on priorities instead of working towards delivering value to the business and its customers (Schwartz, 2016). One study analyses how organisations estimate business value, stating that most respondents tied the business value estimation to their customers, analysing the benefit a single feature would produce for a customer, but also analysing the negative value or impact a feature not delivered can have on a customer (Bakalova et al., 2011). Furthermore, the study explains how other factors, like size and effort estimation of requirements, as well as learning experiences, can affect the value estimation of requirements. Naturally, when prioritising, requirements can be deprioritised if they have potentially high business value, yet they cost a lot to develop (Bakalova et al., 2011).

In one of the previous sections, business value was defined as a set of measurable benefits, usually defined in financial terms (Sward, 2006). This indicates that business value can always be measured, and product owners can conclude that they have achieved high business value if the project results in high financial gain. In contrast, authors argue that high financial gain means high business value. For example, if business value is dependent on both generating revenue and a positive impact on society, then increasing revenue while causing negative effects on society will not only decrease business value but can also be considered as gaining negative value (McGreal & Jocham, 2018, p. 50). Some authors maintain a firm stance on estimation and measurement while analysing the correlation between value measurement and decision-making. Their opinion is that if value cannot be measured, then the organisation cannot manage value properly (Boehm, 2006). Schwartz (2016) believes that the idea of having one business value metric is misleading, as a lot of things constitute business value in an organisation. Some factors are quantifiable, but there are also some that

are not quantifiable at all. Other authors agree with the difficulty of quantifying some factors that define the business value of a requirement; therefore, they suggest the use of relative measures for requirements where value cannot be quantified. So, instead of using exact measures, relative value would be assigned to requirements, which would avoid any difficulty in calculating exact business value yet help product owners compare requirements and prioritise according to business value (Pugh, 2011). Although there are conflicting opinions on whether it is wise and even possible to estimate the business value of requirements, numerous techniques of business value estimation have been found in the literature, which will be explained in the following section. These techniques have been gathered from multiple sources and categorised as per my understanding, due to ease of comprehension and the absence of such categorisation in the literature.

4.2.1 Quantitative techniques of estimating business value

When business value can be quantified and expressed in financial terms, then the financial metrics are suitable for business value estimation to be performed, and the results can feed into the prioritisation of requirements. These metrics mainly focus on the expected income and costs of a project's investment (SCRUMstudy™, 2016). Having this in mind, these techniques are based on the quantitative aspect of business value, part of what has previously been defined as tangible benefits (Project Management Institute, 2019). Here are some of the most well-known financial metrics for estimating business value (Project Management Institute, 2019):

- Return on Investment (hereinafter: ROI)
- Net Present Value (hereinafter: NPV)
- Internal Rate of Return (hereinafter: IRR)
- Payback Period (hereinafter: PBP)
- Benefit-Cost Ratio (hereinafter: BCR)

The following section provides a brief overview of the quantitative techniques of estimating business value. First, there is the ROI, which is an assessment of the expected net income gained from the delivery of a product or a set of features, where expected project revenue and cost are known, calculated as shown in equation (1) (SCRUMstudy™, 2016, p. 74):

$$ROI = \frac{(Project\ Revenue - Project\ Cost)}{Project\ Cost} \quad (1)$$

Oftentimes, ROI is crucial and one of the default metrics used to understand whether an investment or project will increase business value. There has even been an understanding that the estimated increase in business value is proportional to the increase in ROI (Sward, 2006). Even though ROI is a popular financial metric and serves some organisations in estimating business value, it also faces criticism for not being fit for a project in an Agile environment. General disadvantages show that ROI does not take the risk factors into

account and that ROI cannot be calculated for separate requirements, as software requirements are generally intertwined (Schwartz, 2016).

NPV measures the net benefits and investment costs, translated into present value and analysed through the factors of time, risk, and opportunity cost (Schwartz, 2016). Although NPV can give an insightful financial input for decision-making, it is assumed that product owners would find it time-consuming and inefficient to measure NPV at the level of a software requirement in order to prioritise (Schwartz, 2016). The IRR complements the NPV metric by offering another perspective on the analysis of an investment (Sward, 2006). It presents a discount rate at which the present value of incoming revenue equals the present value of outgoing costs, used to evaluate a project's rate of return. And with a greater IRR, an investment is considered to be more financially viable (SCRUMstudy™, 2016).

PBP is a straightforward metric that helps an organisation understand the amount of time needed to earn back the organisation's initial investment, taking into account the financial risk an organisation faces from the perspective of time. Essentially, a bigger PBP means a riskier investment. However, the PBP does not take into account the time value of money, which means it doesn't take into account the changing value of money potentially affected by inflation, risk, etc. (Cohn, 2006). Finally, the BCR defines the ratio of the present value of the benefits of an investment to the present value of costs. The BCR is used to understand if the benefits of an investment outweigh its costs. So, for a ratio greater than 1.0, it can be concluded that benefits outweigh costs, which is a positive indicator. The cost-benefit ratio is a similar metric, where the denominator is the benefit. This means that if the ratio is greater than 1.0, then the costs of that investment are higher than its benefits, meaning that the investment should generally be avoided (Project Management Institute, 2019).

4.2.2 Other techniques of estimating business value

The financial metrics covered in the previous chapter may be well established and structured; however, they are not always applicable to estimate the benefits of an investment. As emphasised by some of the business value definitions, some benefits gained by the organisations cannot be quantified (the intangibles); therefore, product owners may need other techniques to estimate a product's or requirement's business value. Agile practitioners use comparative/relative sizing of each requirement to get to a meaningful estimate, which will indicate the effort, value, cost, or anything they are estimating. The following techniques allow for a requirement to be estimated based on its tangible and intangible benefits (Mallidi & Sharma, 2021):

- T-shirt Size
- Bucket System
- Dot Voting
- Ordering Method

There are studies that enrich this set of techniques with additional ones, such as:

- The Analytic Hierarchy Process (Saaty, 1987)
- Value Point Counting (Mohamed & Wahba, 2008)
- Benefit Points (Hannay et al., 2017)
- Relative Weighting (Wiegers, 1999)
- Kano model and Customer Satisfaction Coefficient (Berger et al., 1993)

The following section provides an overview of the estimation techniques outlined above. The T-shirt Size and Bucket System techniques are recommended to be used when working on a larger backlog with many software requirements. Each requirement can easily be assigned to a certain bucket or assigned a value indicating the category, from smallest to largest. Both these techniques are collaborative, and the goal is to roughly estimate and categorise each requirement so that later the requirements within each bucket can be analysed in more detail (Dinwiddie, 2020). For the Dot Voting technique, all requirements are laid out in a common space for all members taking part in this estimation. Each member is given a limited number of dots that they need to place on one or many requirements from the product backlog, based on their perception of importance. Once they all finish, the dots per requirement are summed up, and the requirements are ordered as per the number of dots gained, from highest to lowest (Rubin, 2012). This is a simple yet effective technique to align with the stakeholders, customers, and end users and validate the importance of each requirement (Patton & Economy, 2014). Then there is the Ordering Method, which is flexible in the use of a scale upon which requirements should be ordered. It focuses more on the factor analysed, in this case, business value, upon which the requirements will be ordered from lowest to highest. Therefore, each requirement is being compared to the others in the backlog and ordered based on its business value in comparison to the others (Mallidi & Sharma, 2021).

AHP is a decision-making method, based on a previously defined comparison scale, where the process is conducted using continuous paired comparisons between items, a process that is known to decrease the level of judgmental errors (Saaty, 1987). The AHP is generally known to be a prioritisation method; however, due to its weighting characteristics, pairwise comparison, and use of relative values, it can also serve as a tool for business value estimation. The way AHP works is that it first requires a goal, alternatives (options), and a hierarchy of criteria and sub-criteria to be defined. The evaluation between the alternatives based on the defined criteria happens through pairwise comparisons. The advantage of this method is that it allows for the criteria and sub-criteria to be analysed through a different level of importance, which is done by assigning weighted scores to them before the final evaluations on alternatives are performed. At the end, the result is a list of alternatives with relative scores that allow for easy prioritisation (Saaty, 1987). Some authors make use of AHP by incorporating it in their own prioritisation method, like the cost-value approach. This approach uses AHP to assess the relative value and relative cost of requirements, which

would then be used to compare and decide on priorities. Ideally, the evaluation of the relative value of each requirement should be done by the customers and users, or someone who would be considered as their suitable substitute, whereas the relative cost would be assessed by the development team (Karlsson & Ryan, 1997).

Many years ago, experts recognised the importance of value estimation of software requirements by emphasising the dynamically changing customer requirements and how the product should adapt to these changes by reshaping the product value. Having this in mind, they proposed a technique that should be useful for Agile projects, where the value gained from a product is compared against its planned budget. These authors came up with a method they called Value Point Counting, consisting of eight steps of identification, categorisation, and calculation. Throughout the steps, the authors are paying attention to analysing the type of requirements but also calculating based on fourteen general requirement characteristics and their degree of influence. Furthermore, they are assigning values to map the importance of each stakeholder for the product/organisation by calculating the Stakeholder Adjustment Factor. This is later used, together with the calculations for the unadjusted value point (UVP) and the value adjusted factor (VAF), to calculate the total value point, which is a relative score upon which prioritisation decisions can be made (Mohamed & Wahba, 2008).

Apart from the already known effort estimation technique of assigning story points, some authors are proposing a similar one, which they call Benefit Points. This method focuses on a scope's epics, as higher-level requirements, and their relation (or contribution) to the overall objectives of the project. The authors suggest the use of relative sizes, like the Fibonacci numbers, as these are sizes Agile practitioners are already familiar with when estimating story points of requirements. The epics and objectives are represented in a matrix, which provides estimated values that can be observed from both the epic and the objective side (Hannay et al., 2017). Overall, the authors are highlighting the benefits of this practice, however they also share their experience, where they've presented the approach to 240 product/project management practitioners and they've stumbled upon some difficulties the practitioners were having with the approach, part of them seeing the difficulty of using the benefit points method for a project with many epics and many objectives (Hannay et al., 2017).

The Relative Weighting is a prioritisation method proposed by Karl Wiegers, aiming to propose a calculation-based priority order of requirements relying on their value, cost, and risk (Wiegers, 1999). Since this is a prioritisation method instead of a business value estimation technique, I will only be analysing a portion focused on calculating the total value of a requirement. The total value for each feature consists of the benefit that a feature provides, but also takes into account the penalty, the negative impact of that feature not being included in the delivery of the scope (Cohn, 2006). The estimation of benefit and penalty is conducted using relative sizes, where a value from 1 to 9 is assigned individually for benefit and penalty; consequently, 1 is the lowest and 9 is the highest value (Wiegers & Beatty, 2013). Once relative values are assigned, the total value for each feature is calculated (by

summing up relative benefit and relative penalty), with the flexibility to even assign relative weight to the benefits and penalties in case one is more important than the other. From here, the value percentage can be calculated, which basically shows the relative contribution of one feature to the value of the whole group of features (Cohn, 2006). Negrão and Guerra (2011), as part of their Benefit-Risk Effective Weighting Model, concluded that the relative weighting is useful when it comes to estimating and prioritising requirements, combined together with the Probability and Impact Matrix. However, they criticise the 1 to 9 scale used for the initial estimation of benefit and penalty and instead propose using the Fibonacci scale in order to get better relative estimates (Negrão & Guerra, 2011). Furthermore, it is suggested that the participants conducting the relative weighting should be the project manager, main customer representatives, and development representatives, where the main customer representatives are the ones deciding upon the benefit and penalty estimate (Wiegers, 1999). Overall, the method is suggested to be used with a smaller set of features in order to be efficient, since the bigger the backlog, the more complex and time-consuming using this method becomes (Wiegers & Beatty, 2013).

The final technique is the Kano Model, which classifies product requirements and features into five categories, based on inputs from the customers. These five categories are: Attractive, Must-be, One-dimensional, Indifferent, and Reverse (Berger et al., 1993). Kano model's classification serves mostly for qualitative data, and when used in isolation, it does not provide any specific quantitative data for analysis. However, Berger et al. (1993) have proposed an extension to the Kano model by adding the element of quantifying customer satisfaction of each product requirement. In essence, customers answer questions for each requirement in terms of how they feel if a feature is present (functional) or absent (dysfunctional) in a product. The answers are then mapped to the Kano categories, where aggregated data is used to calculate the numerical satisfaction coefficient, more specifically, the Satisfaction Index (SI), Dissatisfaction Index (DI), and Average Satisfaction Coefficient (ASC). At the end, each requirement is assigned a numeric output which represents the effect each requirement has on the overall customer satisfaction (Berger et al., 1993). Even though this model does not directly estimate business value, it is still a good customer satisfaction estimation tool, and customer satisfaction is an important component of the business value (Schwartz, 2016). Therefore, this model can be used to get a numeric output of the customer satisfaction of each requirement, which generally can be perceived as a proxy for business value, allowing organisations to estimate the customer value of each requirement and prioritise accordingly.

5 EMPIRICAL RESEARCH ON BUSINESS VALUE ESTIMATION OF SOFTWARE REQUIREMENTS IN THE IT INDUSTRY

The literature review on business value estimation, captured in the previous chapters, shows that there are business value estimation techniques in place, available to be used according

to the organisation's needs. However, as part of my experience working as a product owner in the IT industry, I have rarely seen any product owner explicitly use any kind of technique to estimate the business value of requirements and prioritise the backlog according to it. This triggered my curiosity to research this topic among various product owners from different industries, working in different setups and towards different goals, to potentially find a structured process in which they decide on the priority of requirements. This chapter outlines the research methodology that helped me gather and analyse primary data, which I later compared with the findings from the literature review. It also presents the research results and how they have been analysed and interpreted to become meaningful insights.

5.1 Research methodology

This thesis combines theoretical and empirical research methods, where theoretical methods were used to understand and set a conceptual foundation of the topic, while the empirical research aimed to examine the concepts in practice. For the first part of defining the concepts, identifying the challenges, and analysing how the business value estimation of requirements is approached in literature, I used a literature review, taking into account various books, research papers, and professional articles covering the topic of interest. The findings from the theoretical research helped me set the base and structure of the empirical research. Qualitative research has been chosen for the empirical part due to its contextual insights, flexibility in data collection, in-depth analysis of individual experiences, and interpretivism. The purpose of the qualitative research is to gather and analyse individual experiences and understanding, aiming to find patterns and trends in the way product owners understand business value, whether and how they estimate it as part of their job responsibilities. Consequently, this makes the research of an exploratory nature.

As part of the empirical research, data were collected through observation and in-depth interviews (Appendix 1). This study employs observation as a supplementary method, with varying levels of researcher involvement, aiming to provide contextual insights and support the interpretation of the interview data. Some observations were conducted as non-participant, unstructured observations, focusing on the estimation and prioritisation activities of other product owners in my company, while unstructured self-observation was employed during my role as a product owner in the same company. This combination allows for exposing both perspectives of direct insights gathered from my own experience and external perspectives from the product owners I have observed. Observation was mainly conducted during prioritisation sessions like backlog refinement meetings, sprint planning meetings, or product alignment meetings between product leaders, during 2021. Specific attention was given to the decision-making factors, business value estimation practices, and prioritisation methods used. Data was collected through written notes recorded at the end of each observation session. The data gathered from observation was later analysed along with the interview data gathered from the qualitative research. As for the other part of the qualitative research, I used in-depth interviews with selected product owners from different companies

in the IT industry to gather primary qualitative data. The interviews were conducted during the month of November 2021. The collected data helped me answer my research question: *How do product owners estimate the business value of software requirements in order to effectively prioritise the product backlog?*

The in-depth interviews were structured to follow a well-defined set of open interview questions aimed to be used with all the interviewees. The interviews were based on a semi-structured approach, where participants were guided through the interview questions (Appendix 1), allowing them to express opinions, recall past experiences, and come up with their own definitions. The main interview questions were shaped by themes originating from the literature review, such as:

- The lack of a universal definition for business value (Racheva et al., 2010a) and the vagueness and lack of clarity of existing definitions on business value (Schwartz, 2016). Consequently, this triggered my interest in understanding how product owners perceive and define business value.
- The idea that some product owners prioritise based on intuition and gut-feel (Cohn, 2006). This made me question what is going on in practice, whether product owners are aware and actually make use of any prioritisation method.
- The guidance that prioritisation should happen based on the business value as one of the main factors (Highsmith et al., 2020), which interested me in understanding the factors product owners take into account when prioritising and whether business value is amongst them. And if they consider business value, then to understand how they actually estimate it.

A total of eight participants were interviewed online, through platforms like Microsoft Teams and Slack, during scheduled online meetings that lasted between 30 and 45 minutes. The interviews were recorded and transcribed, so that they can later be analysed individually but also in comparison to one another. Before the interview, each participant had been asked for consent to be recorded, upon which they all agreed. All interviews were conducted in English. The participants had not been informed about the exact title of the thesis and the topic of the research prior to the interview, in order to avoid leading the answers towards the topic of interest, which is business value estimation. The participants in this qualitative research ranged in age from 25 to 45 years old, with an average of 36 years. There was an equal balance of male (50%) and female (50%) participants. As presented in Table 1, the participants came from different industries and countries, including North Macedonia (25%), Serbia (25%), the United Kingdom (12.5%), Sweden (12.5%), Denmark (12.5%), and Germany (12.5%). Furthermore, 75% of the participants have 10-15 years of experience, whereas 25% have 5-10 years of experience as a product owner. It was important to choose participants working explicitly as product owners, as this thesis is analysing the business value estimation from that particular role in the organisation. Overall, it was important to select product owners from different industries, working in different countries and companies, with different seniority levels, and working in either a product or an outsourcing

company setup. For the purpose of this thesis, their identity will be hidden, and they will be referred to as product owner (hereinafter: PO) PO1, PO2, PO3, etc.

Table 1: Summary of demographic and professional background data of participants

Participant	Gender	Country	Industry	Methodology / Framework	Project Management Tool
PO 1	Male	North Macedonia	Health	Agile / Scrum	Jira
PO 2	Female	United Kingdom	Insurance and finance	Agile / Scrum and Kanban	Jira
PO 3	Male	Denmark	Telecommunications	Agile / Scrum	Jira and Azure DevOps
PO 4	Male	Sweden	Retail	Agile / Scrum	Jira
PO 5	Female	North Macedonia	Media and animation	Agile / Scrum	Trello
PO 6	Female	Germany	Civic and social organisations	Agile / Scrum	Jira
PO 7	Male	Serbia	Music	Agile / Scrum	Jira
PO 8	Female	Serbia	Aviation and flight planning	Agile / Scrum	Jira

Source: Own work.

When selecting the participants, the goal was to find product owners working in various industries, in different markets and setups, like outsourcing or product companies. This was important in order to analyse different behaviours that the industries themselves might dictate. Having product owners from different industries as participants in the interviews of this thesis enriches the variety of this research, although the sample per industry is quite small to attempt to find patterns in the product owners' behaviour as per the given industry. The data gathered from the empirical research were analysed using thematic analysis, due to its flexibility and open interpretation of the participants' experiences and understanding of different concepts (Braun & Clarke, 2021). For the codes, I have used an inductive coding strategy. My analysis of the interviews was performed based on the proposed process, as shown in Figure 3, which starts with in-depth familiarisation with the data and ends with results reporting.

Figure 3: Steps in thematic analysis



Source: adapted from Braun and Clarke (2021).

The first phase of the process was marked by data familiarisation, where I spent some time listening to the recorded interviews and reading through the transcripts to get more familiar with the participants' opinions, emotions, and anything that had the potential to be an insight. The second phase revolved around the coding of the transcripts, where initial codes were identified and marked across the valuable text segments of the transcripts. Once the initial code list has been set, the codes have been reviewed to check for any modifications that would capture many codes under one general code. This is where the third phase begins, where codes are analysed and potentially grouped to represent a single theme. Themes are categorised patterns found throughout the dataset. By the end of this phase, a total of ten themes have been identified as valuable patterns found within the research material. In the fourth phase, the initially set themes have been reviewed, associated with the right participants' quotes, and justifications. During this phase, I have identified that some themes are conceptually linked; therefore, I grouped nine of the themes into three overarching themes, each encompassing three closely related sub-themes. The remaining tenth theme remained as a standalone theme. It was important to cluster themes together in order to provide a better overview of the findings. The fifth phase focused on defining the final themes and their sub-themes, as well as their names. The themes were analysed in relevance to the research question. Eventually, the results present one standalone theme and three overarching themes containing three sub-themes each. The results were produced as part of the sixth phase, where the themes are shown in relevance to their sub-themes and participants' quotes, as well as the interpretation of the theme's findings.

Even though a literature review has been conducted prior to the empirical research, the findings from the literature review have no methodological input into the initial coding of the participants' interview data. Instead, the coding originates from the participants' responses and the identified patterns. However, the discussion section will analyse the themes from both empirical research and the literature review perspective.

5.2 Research results

The thematic analysis of the interview data gathered during the qualitative research resulted in four major themes. These results were produced by conducting a thematic analysis process, which consisted of thorough analysis, interpretation, coding, and comparison of the data. Codes have been marked for each valuable text segment in the transcripts (as shown in Figure 2.1, Appendix 2). Upon careful review of the codes, patterns have been identified in the participants' responses, and a total of ten themes have been defined based on the interpretation, similarity, and relevance of the codes (as shown in Figure 3.1, Appendix 3). Due to similarity and shared meanings of the themes, the resulting ten themes were revised and regrouped into four major themes, which are summarised in Table 2. Out of those four themes, one remains as a standalone theme due to its distinct meaning to the research question, whereas the other nine themes were grouped into three overarching umbrella themes, consisting of three sub-themes each.

Table 2: Summary of themes and sub-themes as a result of thematic analysis

Theme type	Theme/Overarching theme	Sub-theme
Standalone Theme	1. Individually constructed understandings of business value	/
Overarching Theme 1	2. Prioritisation as shared responsibility across several roles	2A. Product owner receives decision on initial product backlog priority
		2B. Prioritisation as a joint effort between PO and PM/Business stakeholders
		2C. Product owner and development team provide inputs for reprioritisation
Overarching Theme 2	3. Prioritisation factors and indicators	3A. The customer as main focus of business value and prioritisation
		3B. Main prioritisation factors
		3C. Order in the list of requirements as an initial priority indicator
Overarching Theme 3	4. Prioritisation and business value estimation techniques	4A. Prioritisation and Business value estimation techniques used
		4B. Prioritisation and Business value estimation techniques known but not used
		4C. Prioritisation / Business Value set based on intuition

Source: Own work.

5.2.1 Theme 1: Individually constructed understandings of business value (standalone)

While trying to understand how product owners estimate business value in the software industry, I found it important to understand the perception of product owners and their knowledge of the concept of business value. And in order to understand how they estimate business value, it is important to realise what they are trying to estimate in the first place. Therefore, by analysing the interview transcripts, I have found patterns that led me to conclude that product owners view the business value as a benefit to both the organisation and their customers. *“Business value was always the concept of ...to give the customers enough value so they could upgrade to the new version of the application, meaning that with that they would actually buy the new version and that will bring revenue for the company”* (PO 1). Some product owners also bring the perspective of building the right things for the right people, or as one product owner explained, *“...making sure that we’re building features and functionality that they (customers) need and will add value to their business and in turn help us grow and do more”* (PO 2). Furthermore, the financial gain has been mentioned several times, which means that product owners view the benefit to the organisation initially as a financial benefit. However, some product owners also mention other benefits as well, such as growth and brand image. Overall, the part found as a repeating pattern of business

value definition was that the business value is seen through the prism of customer satisfaction and customer retention. This means that the product owners who were interviewed tend to think that the customer is the most important stakeholder to an organisation.

5.2.2 Theme 2: Prioritisation as shared responsibility across several roles (overarching)

When looking at this theme from the perspective of the research question, it provides context on whether product owners are fully responsible and accountable for the prioritisation process. The conclusions from the empirical research may motivate further research to analyse prioritisation from the perspective of other roles in the organisation.

Sub-theme 2A: Product owner receives decision on initial product backlog priority

Some product owners have mentioned that the initial priority of the product backlog was not done by them personally, but they have inherited it from some other role within the organisation. Generally, this would be a role that deals with the product and the organisation on a higher, more strategic level, like a product manager, a business representative working closely with the customers, etc. *“So, we will do what we class as the epic level prioritisation. I guess in a simplistic format, the epic level would be by the product manager, and it would then be kind of handed over to me to do feature level prioritisation”* (PO 2). However, this does not mean that the product owner has no further permission to reprioritise. It just means that the product owner is working more on the reprioritisation of the sprint backlog, while the overall product backlog may be controlled by a role or group on a higher strategic level in the organisation. This behaviour was also reflected in the observations of the work of product owners, including my own work as a product owner in the same company. All product owners were provided with a rough scope on the product domain they were assigned to work on, which they later analysed, refined, and prioritised.

Sub-theme 2B: Prioritisation as a joint effort between PO and PM/Business stakeholders

In addition to the previous sub-theme, this one focuses on the found patterns in the participants' interviews, discussing the joint effort of the product owner and other roles in the prioritisation process. As previously said, those roles are usually business stakeholders in the organisation with a higher strategic overview and responsibility, working closely with the customers, or sometimes even the customers themselves are the ones working together with the product owner on backlog prioritisation. *“The current setup in the project is that I have equal control over the backlog as the stakeholders on the client side do, so I can say it's like a joint effort in the prioritisation and creating the roadmap for our project”* (PO 7). This statement clearly describes the collaboration between the product owner and the business stakeholders in managing the product roadmap and its prioritisation. Furthermore, prioritisation may fall under the set of responsibilities of the product owner, but that does not mean that the product owner needs to do it alone, or in the words of another product

owner, *“I am the one who is initiating the prioritisation of the backlog, but of course I’m always doing it together with the client”* (PO 8). From the observations, product owners who were observed mainly collaborated with a product manager or product leader, who expects inputs from the product owners in the process of product prioritisation.

Sub-theme 2C: Product owner and development team provide inputs for reprioritisation

Regardless of the role and process used for presenting and passing the initial product backlog priority to the product owner, an interesting pattern was found in most of the participants' answers, which is an activity called reprioritisation. One product owner explains this by saying, *“Once he communicates the priority to me, I then discuss it with the development team and get back to the client with feedback, upon which we might reprioritise”* (PO 5). This means that, even though the product backlog may be initially prioritised by someone else, the development team and their product owner are responsible for reviewing the defined scope and priority and understanding the complexity, resources needed, and feasibility of the set scope and its priority. Another product owner explains this more thoroughly, saying that *“...if the team sees that there are things to be done like the architecture, infrastructure, or refactoring, this is also communicated to the client and let’s say the urgency and importance is communicated so that we can basically have it on the backlog and have the client know why that’s important and do the prioritisation of that as well”* (PO 8). The feedback of the development team and the product owner shapes a process of potential reprioritisation, where the initial priority is reviewed based on the team’s inputs and potentially reprioritised to fit both product and development goals. This pattern has also been identified in the observations, where product owners analyse the scope of work with the development team and provide inputs for reprioritisation based on value, complexity, urgency, dependency, etc.

5.2.3 Theme 3: Prioritisation factors and indicators (overarching)

While searching for ways in which product owners estimate business value in order to effectively prioritise the backlog, it is important to look into the factors and indicators they take into account to prioritise and organise the scope of work. Understanding the factors will potentially shed light on the techniques they use for business value estimation and prioritisation.

Sub-theme 3A: The customer as the main focus of business value and prioritisation

The customer is, undoubtedly, the main pattern in the product owners’ answers when speaking about business value and prioritisation. One product owner defined business value by stating, *“...I would say business value is what we include in order to bring on more customers...”* (PO 2). Another product owner associates the success of the business with customer satisfaction, saying that *“Customer has to be happy - that’s something that is eventually going to impact our business in a positive way”* (PO 4). Yet another product

owner focuses on customers and their effect on business value by bringing the perspective of customer retention, saying that “...*just keeping customers from churning can be seen as a business value*” (PO 3). Overall, these participants' quotes present the customer as the main focus of business value and prioritisation. Customer acquisition, customer satisfaction, and customer retention are just part of the business value factors found as patterns in the empirical research. It is evident that the customer is the main stakeholder, and the success of the customer's business is of high importance to the organisation building the product and their overall business value.

Sub-theme 3B: Main prioritisation factors

The main prioritisation factors that have been found as patterns are business value, customer needs, and the organisation's resources, such as time, effort, team capabilities, knowledge, etc. These are the main factors that product owners have highlighted when discussing their prioritisation practices. “*Business value is definitely one of the key factors taken into consideration when prioritising*” (PO 5). Aside of business value, one of the product owners focuses on the internal resources that affect prioritisation by saying, “...*based on the input from the team, the workload and their speed, I was able to come up with different strategies and different ways of pushing some of the features that we thought will be more demanding in terms of technology, time of development and so on*” (PO 8). Another product owner brings the perspective of end users' problems and their level of importance, suggesting that the bigger and more pressing a problem is for the end user, the higher priority it will have as part of the product backlog. “*Usually, it is the importance of the problem that we want to solve for the end users*” (PO 7). Furthermore, factors can come from external decisions and regulatives, like in the experience of one of the product owners, stating that “*In other cases, business value might be compliance to state regulations that have just been imposed, for security or GDPR purposes and so forth, that is then transferred into business value and prioritised higher than anything else*” (PO 3). Naturally, other factors affect prioritisation as well. Some worth mentioning are the external requirements, like state regulations and security requirements that are mandatory for some organisations. It is important to mention these types of factors since they might not bring any specific value to the customers, yet without them, the organisation cannot perform its basic duties. In addition, a few more factors were identified in the observations, where product owners and product leaders prioritised from the perspective of ongoing technology trends, financial resources, and competitors' strategies.

Sub-theme 3C: Order in the list of requirements as an initial priority indicator

When analysing the backlog, the main indicator that presents the priority of the requirements is the order in which they are listed. This means that requirements with the highest priority are positioned on top, whereas the ones with the lowest priority are positioned at the bottom. “*The main thing is the priority and ordering. So, engineers can see the backlog and say this is the top one - okay, this is the most important one*” (PO 4). This seems to be a general

understanding for all the product owners in the empirical research. Furthermore, they emphasise the understanding of other team members or stakeholders within the organisation to be the same, meaning that order represents the most well-known priority indicator. However, the research is limited in terms of whether the highest priority is equal to the highest business value. Most product owners mentioned that they are not using any additional field to mark the business value and that the order in the list is enough to understand priority. However, there are also examples of product owners mentioning the use of additional fields, aside from the order, to indicate priority, which they use to understand the business value of each requirement. One such example is provided by product owner 3, who said, *“Yes, we actually mark the priority and the business value there (Jira). We have fields on each item in the backlog, where we mark the priority, business value, and complexity”* (PO 3).

5.2.4 Theme 4: Prioritisation and business value estimation techniques (overarching)

Finally, this last theme focuses on specific estimation techniques and approaches that product owners use to effectively prioritise the product backlog. It is an interesting theme as it provides contrasting views on the topic.

Sub-theme 4A: Prioritisation and Business value estimation techniques used

As part of the interviews, there were several techniques that product owners have used, although some of them were not fully aware that they had used a technique at all, or did not know the name of it. Such an example is the quote from product owner 1, where they descriptively explain the dot voting technique their organisation has used to align with customers, but were not fully aware of the technique's name. *“There was one event where they invited the power users of the product from all the hospitals, and then they listed out all the features on a huge whiteboard. And then - all the users were given like 3 stickers, and first there was a discussion to explain what those features were about, and then they just voted with the stickers on whichever features they wanted”* (PO 1). On the other hand, there were product owners clearly stating techniques they have used, which is best represented with the following quote: *“And also the impact versus effort. So that’s something we always try to keep on our minds, because not every big impact feature can be done in a small amount of time, which is expected from us from the stakeholders’ perspective”* (PO 7).

In general, the business value estimation techniques that were mentioned are:

- Dot voting
- T-shirt sizing
- Ordering Method

Whereas, for prioritisation purposes, some product owners have made use of:

- The MoSCoW method
- Value versus effort matrix
- Impact versus complexity matrix
- Impact versus effort matrix

The findings from the observations support the use of the MoSCow method and the impact versus effort matrix as decision-making tools for prioritisation. As for business value estimation, one product owner was observed in their own approach to estimating the business value of requirements. Their approach partially captured the process of value point counting, where requirements were analysed and valued based on different factors and based on the different levels of importance of each stakeholder involved in the voting process.

Sub-theme 4B: Prioritisation and Business value estimation techniques known but not used

Some product owners were able to recall prioritisation and business value estimation techniques they have heard about or used in the past, but are not using in their current projects. One product owner said, *“OH, I thought of another one, that’s more of a prioritisation - the rice model. That’s also used for prioritisation, and it incorporates business value, I think. There’s the Kano model, but I know that is more about the user and what they prefer, which, at the end, I guess it could translate to business value”* (PO 6). The list gathered from the participants’ answers contains several techniques that the product owners have mentioned during the interviews. However, some business value estimation techniques have been mentioned in the context of prioritisation methods and vice versa. Nevertheless, this is a compiled list of all techniques the product owners have mentioned throughout their interviews:

- Return on Investment
- Impact versus effort matrix
- Growth–share (BCG) matrix
- Rice model
- Kano model

Sub-theme 4C: Prioritisation / Business Value set based on intuition

Aside from the techniques used, many product owners stated that they, or their stakeholders, rely on intuition when it comes to assigning business value and prioritisation. One product owner explains the perceived process of prioritisation handled by their stakeholder, saying, *“When discussing the requirements, he makes sure to inform me which requirements are most valuable from his perspective. But I don’t believe that he’s using any estimation technique for this. Maybe...hmm, maybe grouping them as per priority, but again based on his perception of what’s valuable, since I have not been presented with any accompanying data”* (PO5). Another product owner reflects on the experience over the years by saying, *“So I guess from experience over the years, and I think you mentioned gut-feel - a lot of it is*

that for those initial phases” (PO 2). Product owner 3 explained their stakeholders’ decision-making practice by stating, *“So they, I believe, on a hunch or a whim decide...that that Epic might have a higher business value than another one”* (PO 3). “Gut-feel”, “hunch or a whim”, “the business just knows”, all these phrases reveal the pattern of business value definition and prioritisation based on intuition. Even though some product owners have mentioned the use of several techniques, most of the product owners or their stakeholders rely on intuition at some point in the process. This motivates further research to understand whether formal techniques are not needed due to the experience and robust decision-making by product owners or, on the contrary, they should be widely adopted and applied.

6 DISCUSSION

This chapter focuses on an in-depth interpretation of the study’s key findings in the context of the qualitative research and the existing literature. The first part of the chapter focuses on the interpretation of each theme presented in the research results chapter, compared to the findings from the literature review, with respect to my research question. Following this, I will present the limitations of the research. Finally, based on the acknowledged limitations, the last part of this chapter will suggest directions for future research that can cover this topic to a greater extent.

6.1 Interpretation and comparison of key findings

In this part of the thesis, I will give an overview of the previously discussed themes, from both an empirical research perspective and a literature review. The literature review and the empirical research helped me answer my research question, but also opened perspectives for new areas to be analysed and potentially improved. Eventually, these findings should not be taken for granted as they are based on limited information, biased by the experiences of the product owners, and my personal experience as a product owner in the IT industry. This following section will present the interpretation and comparison of key findings as per the respective themes the results are grouped in. Aside of the standalone theme, each of the overarching themes presented below will cover the key findings of all sub-themes contained within.

Theme 1: Individually constructed understandings of business value (standalone)

Understanding how product owners perceive and define business value is inevitable when studying how they estimate business value for the purpose of prioritisation. The general impression from the results is that product owners seem to be aware of what business value indicates; however, when asked to define the concept using their own reasoning, most have faced certain difficulties framing the concept. In this direction, PO 4 has expressed his struggle on the matter by saying, *“Business value...actually that’s a tough one”*. PO 2 has emphasised that she has not used a textbook definition when defining the concept of business

value, clarifying at the end of the definition: “*Not necessarily a textbook definition, but my interpretation of it - a bit of a holistic view*”. This can be compared to the criticism found in the literature, where authors are saying that the concept of business value is vaguely defined and is conceptualised through the context of the latest project the product owner has been assigned to (Racheva et al., 2010a). The latter point is observed in the interviews as well, where product owners try to define business value as seen through their current project, stakeholders’ relationships, and other factors.

Before presenting the insights and proposed definitions, I would like to reflect on my personal impression from this part of the interviews with the product owners. Some interviews, where the participants were previously asked about the prioritisation factors, naturally led to defining the concept of business value since the participants already mentioned the concept alone, which indicates that at least half of them use it regularly as part of their work. However, when asked specifically about a definition, six out of eight product owners struggled to find the right words to begin explaining their views on business value. As seen in the literature review and the empirical research, business value definitions vary; however, some terms are commonly found in both definition streams. The following part will analyse and compare the definitions found in the literature to the definitions resulting from the empirical research.

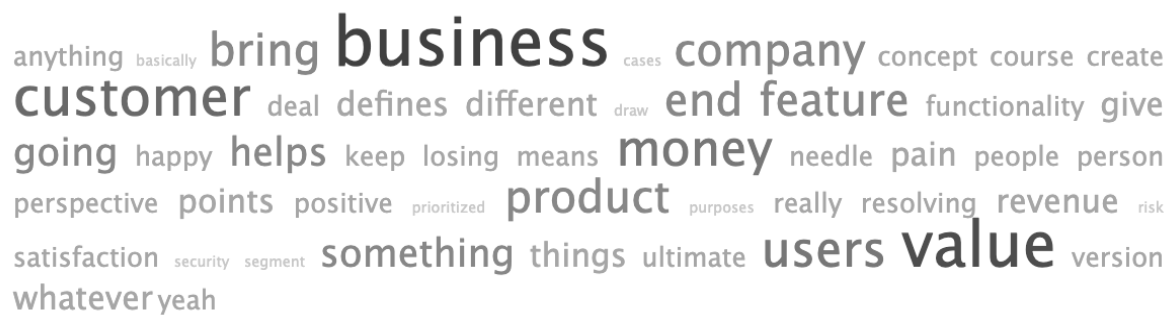
Figure 4: Most commonly used terms within the proposed definitions of business value in the literature review



Source: Own work.

Figure 4 presents the most used terms found in the definitions of business value in the literature, presented by different authors and IT experts. The bigger the terms are represented in the word cloud, the more frequently they are mentioned as part of the definitions. On the other hand, Figure 5 shows the most used terms acknowledged as part of the qualitative research, collected from the definitions provided by the interviewed product owners. This representation of the results helps to easily identify and compare any differences or similarities between the results.

Figure 5: Most commonly used terms within the proposed definitions of business value in qualitative research



Source: Own work.

The initial purpose of the word clouds presented in Figure 4 and Figure 5 is to understand the most used terms in the definitions proposed by the findings of the literature review and the qualitative research. While this does not prove anything contextually, it gives an overview of the overlap of some of the most mentioned terms, aside from business value, like customer, users, benefit, money, feature, revenue, etc. This aligns with the chosen literature, where authors define business value as the willingness and determination to focus on addressing customer needs, which in turn should result in financial gains or other non-financial benefits for the organisation (Harris, 2017). Or simply said in the words of product owner 4: *“Customer has to be happy - that’s something that is eventually going to impact our business in a positive way”*.

Contrary to Sward (2006) and his view on business value only from the financial perspective, the participants have mentioned several intangible benefits they find important when defining business value. For example, PO 3 mentioned the compliance and regulations requirements requested externally, PO 5 mentions the reputation of the organisation, whereas PO 7 even brings the perspective of the development team’s well-being and satisfaction by emphasising *“So it’s a continuous balancing between the happiness of the team and the stakeholders...”*. Therefore, product owners are more inclined to the definitions provided by Schwartz (2016) and the Project Management Institute (2021), which focus on the possibility of quantifying both tangible and intangible benefits.

As a final point, I want to mention how interesting I found PO 3's reaction on business value, where he states the following: *“I’m a product owner, so I actually do not deal with business value. The business deals with business value”*. This comes as opposed to what the literature says that the product owner is responsible for maximising the business value of a product (McGreal & Jocham, 2018). To add to this, Schwartz (2016) is criticising the opinion that the defined business value should come from someone in the business. In fact, he suggests that the whole team needs to be aware of each requirement’s business value, be able to define it, and work towards achieving it.

Theme 2: Prioritisation as shared responsibility across several roles (overarching)

While the literature review reveals insights on the prioritisation process from the product owner's perspective, the empirical research reveals that prioritisation may be a shared responsibility across different roles within an organisation. For seven out of eight participants, prioritisation is a joint effort between the product owner and the product manager, client, or other business stakeholder they collaborate with. In fact, for some of the product owners, the initial list of priorities is provided as initial strategic input of the product backlog, which they have to then refine with the development team and reprioritise as needed. Product managers or other strategic-level business stakeholders prioritise the product backlog, while product owners prioritise the sprint backlog, which is a subset of the product backlog. This can best be seen in the experience of PO 3: *"The change advisory board is mostly involved in the prioritisation of the product backlog. But the sprint backlog - I prioritise that together with the developers on the sprint planning and the refinement..."* This seems to be aligned with part of the literature, where the product manager is the main product-related role that is ultimately responsible for the business value of an organisation; however, it is usually supported by additional roles for any product-related activities such as prioritisation, estimation, and so on (Torres, 2021). Additionally, product managers and product owners meet regularly to discuss and make sure they share the same vision and understanding. This synchronisation provides confidence that the product owner can reprioritise the backlog on their own as they share the same vision and goals with the product manager (Gower, 2013). Unfortunately, synchronisation and well-established collaboration are not required between the product owner and product manager only. As seen in the research results, a product owner usually collaborates with many different stakeholders. If we take PO 2 as an example, she mentions collaborating with a product manager, customers, sales team, solutions team, development team, and so on. However, when specifically referring to backlog prioritisation, PO 2 mentions that the product manager, product owner, and business analyst are all facilitating the process, but on a different level (epic, feature, sprint). It is important that this process is well-defined and organised, as authors are suggesting that the more stakeholders influence the prioritisation process, the more complex the process becomes (Khan et al., 2015).

The empirical research reveals patterns of frequent collaboration between some product owners and their main stakeholders - the customers. In some cases, such as the experience of PO 5 shows, the customer, referred to as the client, is the decision maker behind the product backlog priority. PO 5 says: *"The client mainly did it (prioritisation). Once he communicates the priority to me, I then discuss it with the development team and get back to the client with feedback, upon which we might reprioritise"*. The product owner inherits this decision and works with the development team on further reprioritisation. This aligns with the preference of some authors, who believe that the product backlog should be prioritised by the customers and they should be the ones to decide what the goal of each

iteration/sprint should be (Wiegers & Beatty, 2013). When this is not possible, then the product owner acts as a customer representative and takes over responsibility on their behalf.

Some organisations depend solely on the prioritisation activities of the product owner. PO 4 is part of such an organisation, where he plays the main role in performing and owning the prioritisation of the backlog. This is in line with what is suggested in the Scrum Body of Knowledge, where the product owner is ultimately responsible for managing and prioritising the product backlog. The authors also suggest that the product owner should do this in collaboration with the other stakeholders, by gathering feedback frequently and managing the stakeholders' expectations (SCRUMstudy™, 2016).

When building software, prioritisation is one of the most important activities that can heavily influence the success of the product. Overall, this process helps the development team and the stakeholders understand the reason behind the need and the importance of a software requirement, as well as its urgency to be delivered and used by its customers (Asghar et al., 2016). Additionally, it is quite important to include the right stakeholders in the prioritisation process, as they must possess the right knowledge and skills to assess the requirements compared to the overall goal, as well as be able to find a common ground within the whole group working on prioritising the product backlog. Having a common understanding and building trust between the stakeholders highly affects the success of the prioritisation process (Al-Ta'ani & Razali, 2016).

Theme 3: Prioritisation factors and indicators (overarching)

To understand how product owners prioritise the product backlog, it is important to understand all the factors they take into account when going through this process. Ultimately, the idea was to also understand whether product owners prioritise based on the most common factor, which is business value (Pichler, 2010). As part of the empirical research, a lot of prioritisation factors have been mentioned by the product owners; however, only four out of eight participants mentioned the exact term “business value” or “value” to describe one of the main factors they've based the prioritisation on. The rest of the product owners have not used that particular term, but eventually, what they have described as part of the factors accounts for business value, especially when connecting value to solving customers' pain points, needs, and feedback implementation.

Some authors stress the importance of focusing on customers and their needs instead of focusing only or primarily on business needs. It is fulfilling customers' needs and helping them achieve their goals that helps companies fulfil their business needs. For a product to be perceived as valuable, companies need to be customer-centric (Torres, 2021). Luckily, most of the product owners have distinguished the customer as an important factor when prioritising. Customer satisfaction, customer retention, customer acquisition, as well as transparent and feedback-powered collaboration between the organisation and the customers, are what drive smart and effective prioritisation decisions. Other than the

business value and the customer needs, a common factor mentioned by several product owners was the complexity and the development time, so the overall development effort needed for the implementation of the desired features. PO 4 and PO 6 even mentioned the impact versus complexity or the effort versus value matrix, used to map out all requirements and decide upon their priority. Additionally, factors that were mentioned less but still shape the prioritisation process of different companies are: money earned or saved from losing, regulation, competitors, partners, release dependencies, confidence level, time to market, customer lifetime value, reputation, deadline, etc. If we put these factors against the ones mentioned in the literature, it can be concluded that product owners are focusing on the right factors and are aligned with what the literature recommends. Basically, the priority should be set as a result of the optimal correlation between the following factors (McGreal & Jocham, 2018):

- Business value
- Risk
- Cost/size
- Dependency

Normally, these are not the only factors worth analysing in order to conclude each feature's business value. Some industry specialists often mention the term negative value, which is basically the risks, costs, and consequences of the decision not to develop and deliver a particular feature (Racheva et al., 2010a).

Finally, the order of the PBIs/requirements in a list, usually visualised in a project management tool the team uses, is the main indicator of priority for all product owners part of the empirical research. This is aligned to the literature expectations, where a simple list of requirements ordered per priority results from a technique called relative prioritisation ranking. This technique helps product owners and teams understand the priority of each requirement relative to the others and allows for easy modifications to be made in terms of reprioritisation (SCRUMstudy™, 2016). While the order of requirements is enough to understand the priority for most product owners, PO 3 shares his experience where the order as the main priority indicator is accompanied by additional fields being marked, like a field for priority, business value, and complexity. This is how he explains the process: *“Yes, we actually mark the priority and the business value there. We have fields on each item in the backlog, where we mark the priority, business value, and complexity. The business marks the business value, the developers mark the complexity, and I then mark the priority as a combination of the two”* (PO 3). This organisation uses the Fibonacci sequence as a relative sizing scale to indicate specific values for all three parameters.

Theme 4: Prioritisation and business value estimation techniques (overarching)

This last theme tackles the research question of this thesis from the perspective of identifying product owners' opinions and approaches towards prioritisation and business value

estimation. Previously, in the literature review, two groups of business value estimation techniques have been recognised, divided into qualitative and other techniques. As part of the empirical research, some of the interviewed participants mentioned they use three business estimation techniques, which are essentially techniques that use comparative/relative sizing. The three techniques mentioned are:

- Dot voting
- T-shirt sizing
- Ordering method

Additionally, the participants have shared that they know of but do not necessarily use the business value estimation technique of ROI. One participant mentioned the planning poker technique as a potential technique for business value estimation. When using planning poker, the product owner provides an initial definition and description of the requirement, after which each team member votes individually and privately, assigning the relative estimate they deem adequate. The estimates, once provided, are revealed simultaneously, and discussions within the team are encouraged until a consensus on the estimation is reached (Poženel & Mahnič, 2016). Even though this technique is listed as an Agile estimation technique (Mallidi & Sharma, 2021), it is generally used for estimating the effort needed to implement backlog requirements (Cohn, 2006), instead of estimating their business value. So, instead of revealing the gain of implementing certain requirements, planning poker helps teams estimate and communicate the effort and complexity, therefore the cost of implementing the discussed requirements.

As for prioritisation methods, the MoSCoW method is the one that product owners have used, yet it overlaps with existing techniques from the performed literature review. Additionally, product owners have mentioned the use of different matrices, which help them make better prioritisation decisions, and known, but not used techniques like the Kano model, Rice model, etc. It is worth noting that some product owners may be unaware that they are using any kind of technique, or they do not know its name. This is best seen in PO 7's answer, where he says: *“ I'm probably using a lot of techniques, but I'm not familiar with the actual names, and I see it as doing by intuition, and that is something I am relying on most of the time when I'm prioritising things and working on whatever. So, intuition, but at the end of the day, people tell me that I have been using a technique of which I have not been aware of”* (PO 7).

Even though several techniques have been mentioned, another pattern was found to be more consistent in the interview responses, and that is: an intuition-based decision-making process when it comes to understanding the business value of a requirement and the overall prioritisation. At least half of the participants think that decisions come as a result of previous experience and intuition, believing that the business stakeholder or the domain expert knows best and uses no specific technique to estimate business value or prioritise. For some authors, this is accepted behaviour, as prioritisation does not have to be a formal activity and can rely

on the product owner's intuition (Cohn, 2006). It is believed that a good product owner can prioritise effectively based on intuition and experience, in collaboration with the development team (McGreal & Jocham, 2018). However, there are authors who have a firm stance on measuring business value to avoid any subjectivity and tendencies towards the priority of certain requirements (Racheva et al., 2008). Eventually, some authors criticise most business value estimation techniques and their inefficiency to capture both tangible and intangible benefits (Schwartz, 2016). Others claim that following mathematical formulas will base the decision on an absolute judgement, resulting in product owners strongly believing that they have made the right decision, whereas the idea is to experiment and deliver often to customers, in order to get quick feedback, learn, and reiterate (Torres, 2021). This experimentation mindset is best seen in PO 4's response: *"So it's all about the validation of the value we are creating and how the complexity is...And we're trying to experiment and validate ideas as early as possible, just not going full fetch creating, spending crazy time, and stuff like that. So that's the idea, actually - we're trying to understand the business impact and try to validate the solution if we can do this, and if we do this, this is the right way, and then we just actually say ok. This is the way to go - we validate it, we do the experimentation, whatever is an A/B testing or anything to get to understand if that's the right solution, and then we start building it"*.

6.2 Limitations of the study

This research provides some thought-provoking insights; however, it encounters several limitations. First of all, the sample size of the empirical research is relatively small, which affects the validity of the conclusions. In addition to this, the results cannot be generalised for a whole industry. Second, the interviews were conducted online, which may have affected the engagement of the participants as well as their openness to share experiences. It is important to note that the participants work in IT companies, which may be under a non-disclosure agreement, generally preventing them from sharing too much information of a sensitive nature. The coding part of the data analysis can also be seen as a limitation to this study since it has been done as per my knowledge and interpretation of the participants' answers, which is subject to bias. Inductive thematic analysis supports codes and themes derived from the data itself, which is flexible but also subjective.

This thesis was mainly focused on the work of product owners and their responsibilities around business value estimation and prioritisation. However, as mentioned in Chapter 3.1, Distinction between the product owner and product manager, both the product owner and product manager roles are responsible for the success of the product. Having this in mind, the study is limited to analysing only the product owner's perspective of business value estimation and prioritisation.

Finally, it is important to note that the interviews have been conducted in time just before the widespread use of GAI tools. The current trends suggest that the future work of product

owners is expected to be powered by and supported by GAI tools, which may be used to enable better business value estimation of requirements, and therefore better prioritisation (Gupta et al., 2025). The limitation of this study is that it has not captured the use of any GAI tool in the work of the product owners who have been interviewed.

6.3 Recommendations for future research

Future research can help overcome the limitations of this study and result in further valuable key findings. Initially, further research may focus on expanding the sample size and the variety of the interviewees in terms of their role, the industry they work in, or even the project management tools they use. This way, patterns may occur that are supported with enough data to make conclusions on business value estimation based on the industry, the country of the product owners, etc. Additionally, the interviews can be conducted offline to ensure better engagement and rapport between the interviewer and the interviewee. Improving the comfort and direct communication of the interview improves the richness and quality of the data. Furthermore, future research may include product managers as interviewees, as that role shares some of the product-related responsibilities with the product owner in the company. Eventually, future research should consider using software for qualitative analysis or an AI-supported tool to help with data coding. This way, the researcher's bias and subjective interpretation of data mentioned as part of the limitations will be reduced, and the overall data will be analysed more efficiently.

On the other hand, future research can be inspired by the emerging technologies in the software development industry. More specifically, the current trends in the IT industry are mostly led by artificial intelligence and the efforts to make technology replicate human-like abilities and intelligence (Xi et al., 2023). Focused on the benefit of automating workflows and enhancing decision-making, IT experts are analysing the rise of AI agents, which are autonomous and intelligent systems designed to think and perform actions towards a specifically defined goal (Sapkota et al., 2025). While AI agents can be standalone systems, existing software applications are trying to develop cloud-based, embedded AI agents within their solutions or features that share the characteristics of an AI agent. In this direction, my recommendation for future research is to analyse how a project management tool and its AI agent can influence and potentially improve the business value estimation of product owners, thus helping with efficient and seamless prioritisation.

To support my recommendation and fulfil my goal set for this thesis, I have conducted a brief concept test (Appendix 4), where I investigated business value estimation and backlog prioritisation using the AI agent of a project management tool called ClickUp and its Autopilot Agent. Even though the Autopilot Agent is not completely an AI agent, it is still built based on the main characteristics of an AI agent, which are autonomy, task-specificity, and reactivity (Sapkota et al., 2025). After a quick setup of the software and a brief creation

of a backlog with features representing the requirements of a software application, the following would be possible:

- A product owner could create a business value field for each PBI, set as a number input field, where the business value estimate would be populated.
- A product owner can set up an Autopilot Agent in ClickUp that would work in the background on behalf of the product owner (as shown in Figure 4.1, Appendix 4). It could automate some of the workflows of the product owner and speed up the decision-making process by providing results based on previous data and externally available information.
- A product owner can make the Autopilot Agent assign business value to PBIs (requirements) and reprioritise the backlog accordingly on their behalf (as shown in Figure 4.2, Appendix 4).

Future research can be done in this direction, based on the concept I have presented. Ultimately, the idea behind using an AI-powered project management tool would be to move away from the subjectivity of intuition-based decision-making by the product owner and provide a more objective business value estimation process, which would enable a more automated reprioritisation process by product owners or on their behalf.

7 CONCLUSION

This thesis aimed to understand how product owners estimate the business value of software requirements, with the main goal to prioritise those requirements as part of a product backlog. As the research progressed, additional questions about the definition of business value and the factors of prioritisation emerged during the literature review and were analysed as part of the empirical research. By using interviews and observations for the empirical research, this study aimed to find business value estimation techniques used by product owners in the IT industry and reveal their perception and definition of the term business value, as well as understand their prioritisation practices.

One of the goals of this research was to present a definition of business value. In this regard, this thesis contributed to a better understanding of the concept by providing several definitions, representing the perspective of different authors and product owners. The traditional definitions of business value mainly cover the concept from the financial perspective, calculating the business value through metrics like ROI. However, additional definitions have emerged, suggesting that business value is a benefit not only to the organisation itself but also to its stakeholders, like the customers and their overall satisfaction, as one of the main drivers of business value. This perspective can also be noticed in the answers from the interviews conducted with product owners from different backgrounds. Their definitions of business value include the benefits to the organisation but also to the customers, partners, the public, and other stakeholders. On the other hand, some

authors refuse to define such a broad term as their opinion is that the definition of business value depends highly on the project context, the customer needs, the market position, and other factors. Therefore, even though business value definitions exist, there is still scepticism among authors that business value can be clearly defined, expressing the ambiguity and vagueness of its definition.

Other goals that were successfully attained with this study are the exposure of factors taken into account when prioritising the product backlog and the product owners' tendencies and techniques used when estimating the business value of requirements. Several techniques were presented, although the main findings showed that business value estimation does not seem to be a well-established and structured process that product owners follow. In fact, the interviews have shown that product owners rarely estimate business value and turn to intuition or other decision makers to decide on the business value and, consequently, the priority of software requirements within the backlog. It is interesting to note that some product owners have unknowingly mentioned prioritisation methods as business value estimation techniques, which leads to the thought that there is not enough clarity in the purpose of the methods and techniques that exist. Nevertheless, alongside many factors that were mentioned, most product owners commonly emphasised business value as an important prioritisation factor. Finally, in addition to the analysis of the existing prioritisation practices, this study briefly explored the potential future opportunities of AI-supported prioritisation.

LIST OF KEY LITERATURE

1. Bakalova, Z., Daneva, M., Herrmann, A., & Wieringa, R. (2011). Agile requirements prioritization: What happens in practice and what is described in literature. In D. Berry & X. Franch (Eds.), *Requirements engineering: Foundation for software quality. REFSQ 2011* (Lecture Notes in Computer Science, Vol. 6606, pp. 181–195). Springer. https://doi.org/10.1007/978-3-642-19858-8_18
2. Cohn, M. (2006). *Agile estimating and planning*. Pearson.
3. Dinwiddie, G. (2020). *Software estimation without guessing: Effective planning in an imperfect world*. Pragmatic Bookshelf.
4. Highsmith, J., Luu, P., & Robinson, M. (2020). *Edge value-driven digital transformation*. Addison-Wesley Professional.
5. Mohamed, S. I., & Wahba, A. M. (2008). Value estimation for software product management. In *Proceedings of the 2008 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 2196–2200). IEEE. <https://doi.org/10.1109/IEEM.2008.4738261>
6. Racheva, Z., Daneva, M., Sikkil, K., & Buglione, L. (2010a). Business value is not only dollars – Results from case study research on agile software projects. In M. Ali Babar, M. Vierimaa, & M. Oivo (Eds.), *Product-focused software process improvement. PROFES 2010* (Lecture Notes in Computer Science, Vol. 6156, pp. 131–145). Springer. https://doi.org/10.1007/978-3-642-13792-1_12

7. Schwartz, M. (2016). *The art of business value*. IT Revolution Press.
8. Sward, D. (2006). *Measuring the business value of information technology*. Intel Press.
9. Torres, T. (2021). *Continuous discovery habits*. Product Talk LLC.
10. van Daalen, E., & van Solingen, R. (2024). The current state of operationalizing value by Dutch product owners in agile software development. In D. Šmite, E. Guerra, X. Wang, M. Marchesi, & P. Gregory (Eds.), *Agile processes in software engineering and extreme programming. XP 2024* (Lecture Notes in Business Information Processing, Vol. 512, pp. 89–106). Springer. https://doi.org/10.1007/978-3-031-61154-4_6

REFERENCE LIST

1. Al-Ta'ani, R. H., & Razali, R. (2013). Prioritizing requirements in agile development: A conceptual framework. *Procedia Technology*, 11, 733–739. <https://doi.org/10.1016/j.protcy.2013.12.252>
2. Al-Ta'ani, R. H., & Razali, R. (2016). A framework for requirements prioritisation process in an agile software development environment: Empirical study. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 846–856. <https://doi.org/10.18517/ijaseit.6.6.1375>
3. Asghar, A. R., Bhatti, S. N., Tabassum, A., Sultan, Z., & Abbas, R. (2016). Role of requirements elicitation & prioritization to optimize quality in Scrum agile development. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(12), 300–306. <https://doi.org/10.14569/ijacsa.2016.071239>
4. Asghar, A. R., Bhatti, S. N., Tabassum, A., & Shah, S. A. A. (2017). The impact of analytical assessment of requirements prioritization models: An empirical study. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 8(2), 303–313. <https://doi.org/10.14569/ijacsa.2017.080240>
5. Aurum, A., & Wohlin, C. (Eds.). (2005). *Engineering and managing software requirements*. Springer.
6. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile software development*. Retrieved from <http://agilemanifesto.org/>
7. Berger, C., Blauth, R., Boger, D., Bolster, C., Burchill, G., DuMouchel, W., Pouliot, F., Richter, R., Rubinoff, A., Shen, D., Timko, M., & Walden, D. (1993). Kano's methods for understanding customer-defined quality. *Center for Quality Management Journal*, 2(4), 3–36.
8. Braun, V., & Clarke, V. (2021). *Thematic analysis: A practical guide*. SAGE Publications Ltd.
9. Cagan, M. (2017). *Inspired: How to create tech products customers love* (2nd ed.). Wiley.

10. Campbell, A. (2020). *Agile: Essentials of team and project management*. Alex Campbell.
11. Dooley, J. F. (2017). *Software development: Design and coding – With patterns, debugging, unit testing, and refactoring* (2nd ed.). Apress.
12. Farley, D. (2021). *Modern software engineering: Doing what works to build better software faster*. Addison-Wesley Professional.
13. Gambo, I. P., Ikono, R. N., Achimugu, P. O., & Olaronke, I. G. (2018). A ranking model for software requirements prioritization during requirements engineering: A case study. *International Journal of Computer Science and Information Security*, 16(4), 255–268.
14. Gower, B. (2013). *Agile business: A leader's guide to harnessing complexity*. Rally Software.
15. Gupta, A., David, J., Golli, A., & Sai, V. (2025). Generative AI in the software development lifecycle (SDLC). *Global Journal of Engineering and Technology*, 4(1), 26–31.
16. Hannay, E. J., Benestad, H. C., & Strand, K. (2017). Benefit points: The best part of the story. *IEEE Software*, 34(3), 73–85. <https://doi.org/10.1109/MS.2017.66>
17. Harris, M. D. S. (2017). *The business value of software*. Auerbach Publications.
18. Hron, M., & Obwegeser, N. (2022). Why and how is Scrum being adapted in practice: A systematic review. *Journal of Systems and Software*, 183, 111110. <https://doi.org/10.1016/j.jss.2021.111110>
19. Institute of Electrical and Electronics Engineers. (1990). *IEEE standard glossary of software engineering terminology (IEEE Std 610.12-1990)*. IEEE.
20. International Institute of Business Analysis. (2015). *A guide to the business analysis body of knowledge (BABOK® Guide)* (3rd ed.). International Institute of Business Analysis.
21. Karlsson, J., & Ryan, K. (1997). A cost-value approach for prioritizing requirements. *IEEE Software*, 14(5), 67–74. <https://doi.org/10.1109/52.605933>
22. Kelly, A. (2019). *The art of agile product ownership: A guide for product managers, business analysts, and entrepreneurs*. Apress.
23. Khan, J. A., Rehman, I. U., Khan, Y. H., Khan, I. J., & Rashid, S. (2015). Comparison of requirement prioritization techniques to find best prioritization technique. *International Journal of Modern Education and Computer Science (IJMECS)*, 7(11), 53–59. <https://doi.org/10.5815/ijmecs.2015.11.06>
24. Kuhrmann, M., Münch, J., Richardson, I., Rausch, A., & Zhang, H. (2016). *Managing software process evolution: Traditional, agile and beyond*. Springer.
25. Kukreja, N., Payyavula, S. S., Boehm, B., & Padmanabhuni, S. (2013). Value-based requirements prioritization: Usage experiences. *Procedia Computer Science*, 16, 806–813. <https://doi.org/10.1016/j.procs.2013.01.084>
26. Larman, C., & Vodde, B. (2016). *Large-scale Scrum: More with LeSS*. Addison-Wesley Professional.
27. Lehtola, L., Kauppinen, M., & Kujala, S. (2004). Requirements prioritization challenges in practice. In F. Bomarius & H. Iida (Eds.), *Product focused software process improvement. PROFES 2004* (Lecture Notes in Computer Science, Vol. 3009, pp. 497–508). https://doi.org/10.1007/978-3-540-24659-6_36

28. Lohan, G., Lang, M., & Conboy, K. (2011). Having a customer focus in agile software development. In J. Pokorny et al. (Eds.), *Information systems development* (pp. 441–453). https://doi.org/10.1007/978-1-4419-9790-6_35
29. Lukassen, C., & Schuurman, R. (2023). *Practical product management for product owners*. Addison-Wesley Professional.
30. Magnussen, E. F., Havnelid, E., & Molléri, J. S. (2024). The product owner and its impact on success and challenges in agile scrum projects: An interview study. *Procedia Computer Science*, 239, 1157–1164. <https://doi.org/10.1016/j.procs.2024.06.282>
31. Mallidi, R. K., & Sharma, M. (2021). Study on agile story point estimation techniques and challenges. *International Journal of Computer Applications in Technology*, 174(13), 9–14. <https://doi.org/10.5120/ijca2021921014>
32. McGreal, D., & Jocham, R. (2018). *The professional product owner: Leveraging Scrum as a Competitive Advantage*. Addison-Wesley Professional.
33. Mendonça dos Santos, R. B., Figueiredo, P. S., & Marques, F. T. (2023). Challenges to agile software project management practices in the context of the COVID-19 pandemic. *Gestão & Produção*, 30. <https://doi.org/10.1590/1806-9649-2022v29e9722>
34. Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: A comparative analysis. *International Journal of System Assurance Engineering and Management*, 14(6), 1504-1522. <https://doi.org/10.1007/s13198-023-01958-5>
35. Negrão, E. C., & Guerra, E. (2011). A case study for prioritizing features in environments with multiple stakeholders. In *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion* (OOPSLA '11, pp. 131–140). ACM. <https://doi.org/10.1145/2048147.2048187>
36. Niva, P. (2021). *Freedom in the role of a product owner: A case study* (master's thesis). Lappeenranta–Lahti University of Technology LUT
37. Osterwalder, A., & Pigneur, Y. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers*. Wiley.
38. O'Toole, C., Schneider, J., Smaje, K., & LaBerge, L. (2020, October 5). *How COVID-19 has pushed companies over the technology tipping point—and transformed business forever*. McKinsey & Company. <https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and-transformed-business-forever>
39. Patton, J., & Economy, P. (2014). *User story mapping: Discover the whole story, build the right product*. O'Reilly Media.
40. Pichler, R. (2010). *Agile product management with scrum: Creating products that customers love* (1st ed.). Addison-Wesley.
41. Poženel, M., & Mahnič, V. (2016). Studying agile software estimation techniques: The design of an empirical study with students. *Global Journal of Engineering Education*, 18(2), 53–58.

42. Pressman, R. S. (2000). *Software engineering: A practitioner's approach* (5th ed.). McGraw-Hill.
43. Project Management Institute. (2019). *A guide to the project management body of knowledge (PMBOK® Guide)* (6th ed.). Project Management Institute.
44. Project Management Institute. (2021). *A guide to the project management body of knowledge (PMBOK® Guide)* (7th ed.). Project Management Institute.
45. Pugh, K. (2011). *Lean-Agile acceptance test-driven development: Better software through collaboration*. Addison-Wesley.
46. Racheva, Z., Daneva, M., & Buglione, L. (2008). Supporting the dynamic reprioritization of requirements. In *2008 Second International Workshop on Software Product Management* (pp. 49–58). IEEE. <https://doi.org/10.1109/iwspm.2008.7>
47. Racheva, Z., Daneva, M., & Sikkell, K. (2009). Value creation by agile projects: Methodology or mystery? In F. Bomarius, M. Oivo, P. Jaring, & P. Abrahamsson (Eds.), *Product-focused software process improvement. PROFES 2009* (Lecture Notes in Business Information Processing, Vol. 32, pp. 141-155). Springer. https://doi.org/10.1007/978-3-642-02152-7_12
48. Racheva, Z., Daneva, M., Herrmann, A., & Wieringa, R. J. (2010b). A conceptual model and process for client-driven agile requirements prioritization. In *2010 Fourth International Conference on Research Challenges in Information Science (RCIS)* (pp. 287–298). IEEE. <https://doi.org/10.1109/RCIS.2010.5507388>
49. Regnell, B., Paech, B., Aurum, A., Wohlin, C., Dutoit, A., & Natt och Dag, J. (2001). Requirements mean decisions! – Research issues for understanding and supporting decision-making in requirements engineering. In *Proceedings of the First Swedish Conference on Software Engineering Research and Practice (SERP'01)* (pp. 49–52). Blekinge Institute of Technology.
50. Rida, A. R., Nazir, S. N., Tabassum, A., & Shah, S. A. A. (2017). The impact of analytical assessment of requirements prioritization models: An empirical study. *International Journal of Advanced Computer Science and Applications*, 8(2), 303–313. <https://doi.org/10.14569/IJACSA.2017.080240>
51. Rigby, D., Sutherland, J., & Noble, A. (2018). Agile at scale. *Harvard Business Review*, 96(3), 88–96.
52. Rubin, K. S. (2012). *Essential scrum: A practical guide to the most popular agile process* (1st ed.). Addison-Wesley Professional.
53. Saaty, R. W. (1987). The analytic hierarchy process. *Mathematical Modelling*, 9(3–5), 161–176. [https://doi.org/10.1016/0270-0255\(87\)90473-8](https://doi.org/10.1016/0270-0255(87)90473-8)
54. Sapkota, R., Roumeliotis, K. I., & Karkee, M. (2025). *AI agents vs. agentic AI: A conceptual taxonomy, applications and challenges*. arXiv. <https://arxiv.org/abs/2505.10468>
55. Schlenker, L., & Matcham, A. (2005). *The effective organization: The nuts and bolts of business value*. Wiley.
56. Schwaber, K., & Sutherland, J. (2020). *The scrum guide: The definitive guide to Scrum: The rules of the game*. <https://scrumguides.org/scrum-guide.html>

57. SCRUMstudy™. (2016). *A guide to the scrum body of knowledge (SBOK™ Guide)* (3rd ed.). VMEdu, Inc.
58. Scrum Alliance. (2017). State of Scrum 2017-2018. <https://www.scrumalliance.org/>
59. Sommerville, I., & Sawyer, P. (1997). *Requirements engineering: A good practice guide*. Wiley.
60. Sommerville, I. (2015). *Software engineering* (10th ed.). Addison-Wesley.
61. Szalvay, V. (2004). *An introduction to agile software*. Danube Technologies, Inc. https://profs.info.uaic.ro/adrian.iftene/Licenta/Documentatie/Intro_to_Agile.pdf
62. Thompson, R. G. (2014). *Hooked on customers: The five habits of legendary customer-centric companies*. CreateSpace Independent Publishing Platform
63. Trieflinger, S., Münch, J., Bogazköy, E., Eißler, P., Schneider, J., & Roling, B. (2021). How to prioritize your product roadmap when everything feels important: A grey literature review. In *2021 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (pp. 1–9). IEEE. <https://doi.org/10.1109/ICE/ITMC52061.2021.9570243>
64. VersionOne. (2023). State of agile report (17th ed.). VersionOne. <https://www.stateofagile.com/>
65. Wiegers, K.E. (1999). First thing first: Prioritizing requirements. *Software Development*, 7, 48-53.
66. Wiegers, K., & Beatty, J. (2013). *Software requirements* (3rd ed.). Microsoft Press.
67. Wiegers, K. (2021). *Software Development Pearls: Lessons from Fifty Years of Software Experience* (1st ed.). Addison-Wesley Professional.
68. Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., ... Gui, T. (2023). The rise and potential of large language model based agents: A survey. arXiv. <https://doi.org/10.48550/arXiv.2309.07864>
69. Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In A. Aurum & C. Wohlin (Eds.), *Engineering and managing software requirements* (pp. 19–46). Springer. https://doi.org/10.1007/3-540-28244-0_2
70. Žužek, T., Gosar, Ž., Kušar, J., & Berlec, T. (2020). Adopting agile project management practices in non-software SMEs: A case study of a slovenian medium-sized manufacturing company. *Sustainability*, 12(21), 9245. <https://doi.org/10.3390/su12219245>

APPENDICES

Appendix 1: Interview questions

Below is a list of questions used during the in-depth interviews with the selected product owners. These questions were modified based on the conversations with the interviewees.

1. What's the role within your company/organisation?
2. Which industry are you working in?
3. What is the team size that you are working in?
4. Other than your team, which are the stakeholders you are collaborating with?
5. What is the project management methodology/framework you are currently using?
6. What is the project management tool you are currently using?
7. Who is responsible for the prioritisation process within your company/organisation?
8. What is the priority of the requirements within the product backlog usually based on?
Which parameters are being used?
9. How would you define the concept of business value?
10. Feel free to think about it for a second, think of your own definition of business value, or tell me a definition you have read and find useful, or give me an example of what business value represents from your perspective.
11. Do you pay attention to business value when prioritising the product backlog?
12. How do you estimate a requirement's business value?
13. Hypothetically speaking: if you take a look at your product backlog, what would indicate the amount of business value of a requirement? Or of one requirement compared to another?
14. Which business value estimation techniques can you think of?

Appendix 2: An extract from a coded interview transcript

Figure 2. 1: An extract from a coded interview transcript with the suitable codes assigned

The screenshot shows a 'TA Coding' application interface. On the left, a list of codes is visible, with '3' selected. The main area displays a transcript of an interview. Several segments of the transcript are highlighted in yellow, indicating they have been coded. To the right of the transcript, a list of codes is assigned to the highlighted segments. The codes are:

- Business value as the main focus of prioritization
- Business value as earned revenue, decreased loss and customer satisfaction
- Customer needs as factor in prioritization
- Regulatory requirements as factor in prioritization
- Competitive advantage as factor in prioritization
- Partner company needs as factor in prioritization

Source: Own work.

Appendix 3: Excerpt from the process of grouping codes into identified themes as a step in the conducted thematic analysis

Figure 3. 2: Excerpt from the collected themes assigned to the specific groups of codes

The screenshot shows a 'Codes and themes' application interface. It displays a table with two main columns: 'Code' and 'Theme'. The table lists 21 codes and their corresponding themes. The themes are grouped into several categories, each represented by a different color. The categories are:

- Individually constructed understandings of business value (light blue)
- The customer as main focus to business value and prioritization (green)
- Product owner and development team provide inputs for REprioritization (blue)
- Prioritization and Business value estimation techniques used (red)
- Prioritization as joint effort between PO and PM/Business stakeholders (orange)
- Main prioritization factors (light blue)
- Prioritization / Business Value set based on intuition (light blue)

Source: Own work.

Appendix 4: Automated agent configuration for business value estimation and prioritisation – concept testing

For the purpose of analysing the influence of AI agents on the process of business value estimation and prioritisation, I have conducted a concept test by using ClickUp's AI agent feature, called the Autopilot Agent. With a quick setup of the software and a brief creation of a backlog with features of a hypothetical Restaurant software application, I have successfully set up a simple workspace that has allowed me to test the Autopilot Agent capabilities and come to the following observations:

ClickUp allows for custom fields to be created for each requirement

This is important as it allows for easy data interpretation later on in the process. The software allowed me to create a Business value field, which I have set as a number input field, where the business value estimate would be populated.

A product owner can set up an Autopilot Agent in ClickUp that will work in the background, on behalf of the product owner

The purpose of the AI agent in this sense is to automate some of the workflows of the product owner and speed up the decision-making process by providing results based on previous data and externally available information. Although it is important to note that the current version of Autopilot Agents cannot browse the internet for information.

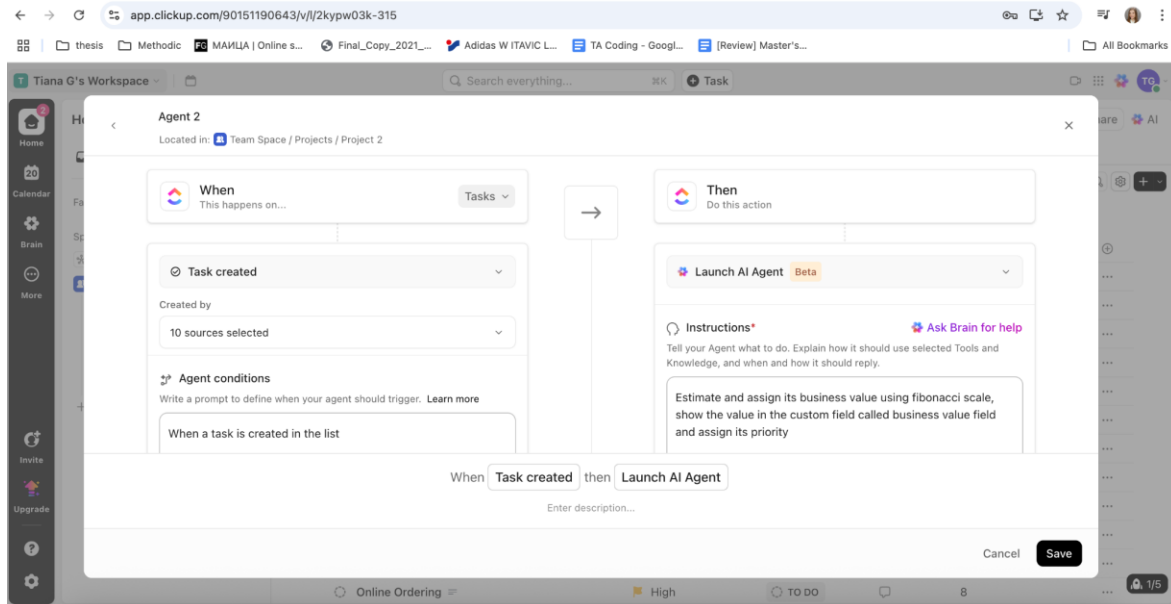
A product owner can make the Autopilot Agent assign business value to tasks (requirements) and (re)prioritise the backlog accordingly

Using a simple prompt, the Autopilot Agent can be set with a simple When - Then rule, defining the conditions under which it has to receive a request and react to it. With some simple testing, I have created an Autopilot Agent that estimates the business value of each new PBI I create and assigns it a priority value (from a given category). Initially, the Autopilot Agent works based on common knowledge on which it is trained; however, with time, the product owner can train the Autopilot Agent according to their needs and preferences. This means that the product owner can set the estimation scale, provide examples, make corrections, and expect the Autopilot Agent to learn from them, ask for an estimation of business value to be executed using a specific methodology, etc.

Naturally, this is just an idea for further research in an area yet to be discovered and experimented with. There are already initial limitations to this research, like the unknown factors that affect the thinking or processing of the Automated Agent, the factors affecting its decisions, as well as the biased results based on the prompts of the product owner. Additionally, this observation was conducted on a product backlog with items that contain only a title and a short description, so these are the only input factors that the Automated

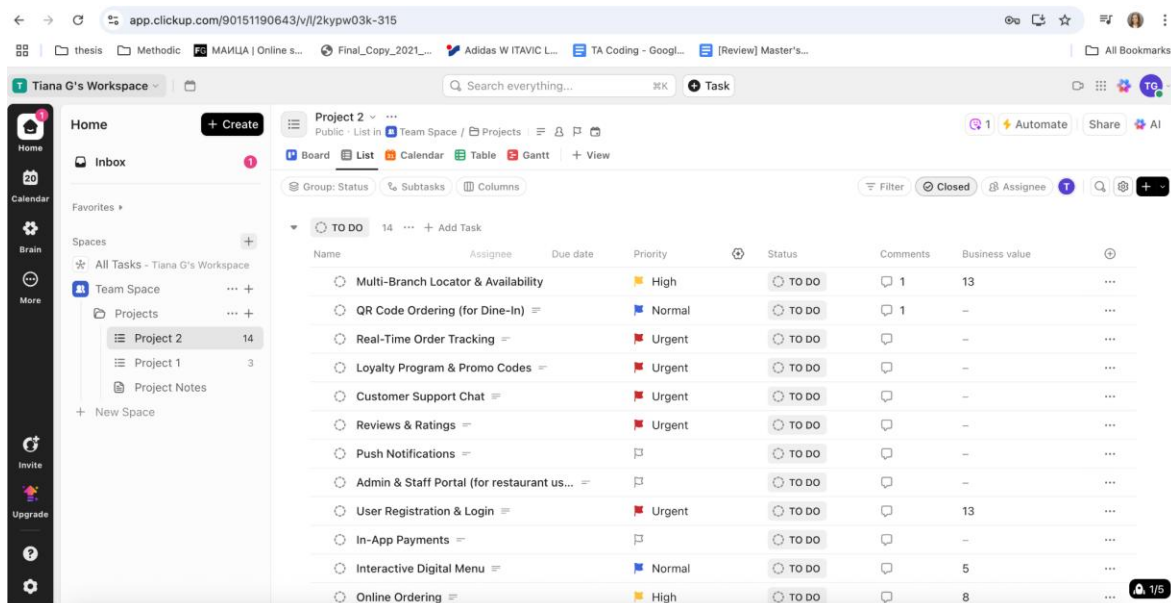
Agent could analyse. However, it may serve as an interesting area for further research and improvement.

Figure 4. 3: Prompt that will train the Autopilot Agent to help the product owner in business value estimation



Source: Own work.

Figure 4. 4: Estimated and prioritised backlog as a result of the automated work by the Automated Agent in ClickUp



Source: Own work.