



**A PROCESS MODEL FOR INITIAL PRODUCT STRATEGY DEVELOPMENT
FOR B2B SAAS**

Supporting productization with state-of-the-art product management

Lappeenranta–Lahti University of Technology LUT

Master's Programme in Software Product Management and Business

2022

Bogdan Moroz

Examiners and supervisors: Associate Professor Sami Hyrynsalmi

Post-doctoral Researcher Andrey Saltan

ABSTRACT

Lappeenranta–Lahti University of Technology LUT
Your school: LUT School of Engineering Science
Your degree programme: Engineering / Software Engineering

Bogdan Moroz

A process model for initial product strategy development for B2B SaaS. Supporting productization with state-of-the-art product management

Master's thesis

2022

139 pages, 44 figures, 9 tables and 1 appendix

Examiners and supervisors: Associate Professor Sami Hyrynsalmi, Post-doctoral Researcher Andrey Saltan

Keywords: product management, product strategy, innovation, software industry, business-to-business, productization

The novel part of this study is a process model for initial product strategy development for B2B SaaS that supports companies productizing their customer-specific software. The study follows the design science research approach by Johannesson and Perjons (2021) and aspires to contribute to both theory and practice by providing prescriptive design knowledge to product managers in charge of productizing customer-specific systems into B2B SaaS. The process model describes the key questions, tools and step-by-step instructions for each of its 6 phases. The process model was demonstrated at a Finnish B2B software company, and led to a detailed product strategy that created a shared understanding of the product. The strategy was approved by the company and will guide further product development. The model was commended by the management as a well-thought-out, well-arranged, and professional take on the process. The study found that the B2B SaaS context of the product influences the product strategy at least in the delivery model, tailorability, pricing, legal, and performance management aspects. The productization context influences at least the positioning and product definition, pricing, sourcing, and financial management elements. The linear structure of the model has been pointed out as a limitation to its generalizability. The study suggests a possible future variation of the model emphasizing the iterative nature of new product development. The study calls for further development and testing of the process model, and continuous refinement of its phases. In the sphere of pricing and financial management, a cross-disciplinary study with scholars of management and finance can further advance the process model. A comprehensive toolkit for product strategy development can also be created.

ACKNOWLEDGEMENTS

I would like to thank my supervisors Sami Hyrynsalmi and Andrey Saltan for their continued support and advice during the thesis process. I would also like to thank Hans-Bernd Kittlaus for kindly providing excerpts from the 2022 edition of his ISPMA-compatible study guide and handbook before its publication, and for sharing his feedback on the process model. Finally, I want to thank my employers and colleagues, who generously participated in the strategy development process, contributed their expertise and feedback, and helped me better understand the inner workings of our company and market.

ABBREVIATIONS

B2B	Business-to-business
CBRN	Chemical, biological, radiological, and nuclear
DPA	Data processing agreement
ISPMA	International Software Product Management Association
SaaS	Software-as-a-Service
SLA	Service-level agreement
SPM	Software Product Management
SPMBOK	Software Product Management Body of Knowledge

Table of contents

Abstract

Acknowledgements

Abbreviations

1.	Introduction	7
1.1.	Objective and research questions.....	8
1.2.	Scope.....	9
1.3.	Method.....	11
1.4.	Structure.....	11
2.	Background	13
2.1.	Software process improvement.....	13
2.2.	Productization of customer-specific software.....	14
2.3.	Business-to-business Software-as-a-Service	16
2.4.	Product strategy	18
2.5.	The ISMPA SPM framework and the SPMBOK	19
2.5.1.	Positioning and product definition.....	21
2.5.2.	Delivery model and service strategy.....	23
2.5.3.	Ecosystem management.....	24
2.5.4.	Sourcing	24
2.5.5.	Pricing	24
2.5.6.	Financial management	26
2.5.7.	Legal and IPR management.....	27
2.5.8.	Performance and risk management.....	29
2.6.	Documentation of the product strategy.....	34
2.7.	Software product strategy in B2B and SaaS context	36
2.8.	Selected tooling	38
2.8.1.	Problem and position statement template	38
2.8.2.	Product Vision Board	39
2.8.3.	Strategy Canvas and ERRC grid from the Blue Ocean family of tools.....	40
2.8.4.	Domain model	43
2.8.5.	Business Process Model and Notation (BPMN).....	44

2.8.6.	Accion pricing framework	45
2.8.7.	Balanced Product Scorecard	50
3.	Design research problem and artefact requirements	52
3.1.	Explicating the problem.....	55
3.2.	Defining artefact requirements	55
4.	Model for initial strategy development for a B2B SaaS product	57
4.1.	Overview of the process model	57
4.2.	Phases of the process model	61
4.2.1.	Collaborative strategy workshop	62
4.2.2.	Product definition refinement	63
4.2.3.	Architecture design	66
4.2.4.	Financial discussion.....	68
4.2.5.	Legal review	70
4.2.6.	KPI selection & risk analysis.....	73
5.	Model demonstration: creating a strategy for Perception Cloud.....	76
5.1.	A brief history of the Perception system	76
5.2.	Strategy design activities with key stakeholders	79
5.2.1.	Collaborative strategy workshop	79
5.2.2.	Product definition refinement	82
5.2.3.	Architecture design	88
5.2.4.	Financial discussion.....	97
5.2.5.	Legal review	104
5.2.6.	KPI selection & risk analysis.....	110
6.	Model evaluation	114
6.1.	Evaluation by observation	116
6.2.	Evaluation by questionnaire.....	118
6.3.	Feedback by Hans-Bernd Kittlaus	119
7.	Discussion	121
8.	Conclusion.....	131
	References.....	134

Appendices

Appendix 1. Sample Key Performance Indicators by Roman Pichler (2016)

1. Introduction

Inspired by the success of large software vendors, a growing number of software companies seek to productize their customer-specific software (Artz et al., 2010). Productization refers to the transformation of software tailored to the needs of specific customers into a standard software product (Artz et al., 2010). Kittlaus and Fricker (2017) define a product as “a combination of goods and services, which a supplier/development organization combines in support of its commercial interests to transfer defined rights to a customer”. A software product is “a product whose primary component is software” (Kittlaus and Fricker, 2017). The shift from customer-specific software to a standard software product calls for increased attention to the product management activities within the development organization (Yrjönkoski, 2018).

But what are the software product management activities? Multiple frameworks exist to answer that question. Paajoki (2020) conducted a literature review and concluded that the Software Product Management (SPM) framework by the International Software Product Management Association (ISPMA) provides the most balanced and holistic approach to outlining the practices and principles of product management. Ebert and Brinkkemper (2014) write that over the past few years the Software Product Management Body of Knowledge (SPMBOK) has been created based on findings from industry and academia. The ISPMA SPM framework serves as the underlying knowledge area framework of the SPMBOK (Ebert and Brinkkemper, 2014). The SPM framework described in ISPMA literature (ISPMA, 2021b) splits the core product management activities into product strategy and product planning. The product strategy defines how the product should evolve over a certain time frame (often 1 to 5 years, varying based on industry). Product planning converts the strategy into an executable plan that a product team can follow day to day (ISPMA, 2021b).

This thesis work is commissioned by a Finnish B2B software company. The company is a global supplier of situational awareness software (SAS), and sells a proprietary software system called Perception (the name is changed for the thesis). Perception is a modular system that supports custom configurations for a variety of use cases, such as vehicle installations, shelter monitoring and control, as well as integration of portable detectors via a mobile app.

The core functionality includes visualizing real-time measurements of CBRN (chemical, biological, radiological, and nuclear) detectors, generating alarms when dangerous limits are exceeded, and providing measurement reports over periods of time. Until this point, the system has been offered to customers as standalone installations in customer-controlled environments. For each customer, the system was customized to an extent. The commissioner company is looking for a way to productize the Perception system and offer it as a SaaS.

The commissioner company aims to position the SaaS version of Perception as a new product. For that product, a strategy needs to be developed. Customers purchasing the Perception system vary in the degree of openness, and some of the customers have stricter privacy and security constraints than others. Some of the customers require an on-premises controlled server infrastructure, and for those customers the SaaS version of Perception would not be suitable. It is necessary to clarify the customer segments for the new SaaS offering.

The Perception software is not offered to individual consumers, but rather to industries and governmental organizations. It is therefore classified as business-to-business (B2B). The B2B product development is characterized as complex by Salameh and Bass (2019). The complexity stems from multiple stakeholders from many business sides involved in decision-making. The parties may have conflicting agendas, priorities, and politics (Salameh and Bass, 2019).

1.1. Objective and research questions

The goal of this thesis is to **design a process that produces a comprehensive product strategy for a company undergoing productization of customer-specific software into B2B SaaS**. The term “comprehensive” is used to imply that the product strategy covers all the 8 elements of product strategy defined by the ISPMA SPM framework. The 8 elements are introduced in detail in Chapter 2. This goal allows the thesis to achieve two objectives. Firstly, the thesis aims to solve a real business problem by demonstrating the process model and producing a product strategy for the SaaS version of Perception. This is the main objective and the driver for the creation of the thesis work. Secondly, the thesis aspires to contribute to the SPMBOK by producing a process model that may be adopted by other

companies wishing to productize their customer-specific software into B2B SaaS. To achieve the stated goal, the thesis answers the following research questions:

- **RQ1:** What process can a company follow to produce a comprehensive initial product strategy?

This is the main research question of the thesis. The ISPMA SPM framework provides a list of aspects that need to be considered to develop a comprehensive product strategy. The framework does not recommend a specific process to perform strategy-related activities, and only suggests executing them periodically, or when triggered by certain circumstances (Kittlaus and Fricker, 2017, pp. 116-117; Paajoki, 2020, p.38).

- **RQ2:** How does the B2B SaaS context influence product strategy decisions?

The product in question for this thesis is a B2B SaaS. The ISPMA SPM framework has a broader scope, meaning it applies to all kinds of software, including SaaS, licenced software products, embedded software, etc. The thesis considers to what extent certain product strategy decisions are influenced by the B2B SaaS nature of the software product.

- **RQ3:** How does the productization context influence product strategy decisions?

The business problem addressed by this thesis work is the productization of an existing software system into a generalized product offering. The thesis discusses to what extent certain product strategy decisions are influenced by the characteristics of the existing software system and the productization context.

1.2. Scope

The thesis focuses on product strategy design for a B2B SaaS product, which is a productized version of an existing customer-specific software system. The thesis focuses on the product management perspective on productization. A separate thesis work at the commissioner company considers the technical implementation of the productization (see Rantanen, 2022). The thesis operates within the theoretical framework of the Software Product Management Body of Knowledge. The thesis uses the ISPMA SPM framework to define the elements that comprise a software product strategy. The ISPMA SPM framework is used because the framework is the foundational knowledge area framework of the SPMBOK (Ebert and Brinkkemper, 2014). Additionally, research by Paajoki (2020) shows that it is the most

balanced and holistic of multiple product management frameworks, focused purely on SPM. Other product management frameworks were not considered in this thesis. While no framework can replace good judgement, it can help avoid common pitfalls and provide structure to experiment, think about, and improve the product (Bavaro and McDowell, 2021, p. 19). The thesis considers the 8 elements of product strategy provided by the SPM framework as comprehensive and complete. The 8 elements are as follows:

1. Positioning and product definition
2. Delivery model and service strategy
3. Ecosystem management
4. Sourcing
5. Pricing
6. Financial management
7. Legal and IPR management
8. Performance and risk management

The 8 elements represent each row of the “Product Strategy” column of the ISPMA SPM framework (ISPMA, 2021a). The goal of the thesis is to develop a process that produces a strategy that covers all these aspects. The definitions and fundamental concepts related to each of these aspects are presented in Chapter 2. Each of these elements can be researched in depth, and entire dissertations were written, for example, on the subject of pricing alone (see Saltan, 2021). To constrain the scope and timeframe of the work, the thesis focuses on providing definitions and key concepts related to each of these elements, based on academic and practitioner publications about SPM.

The thesis proposes a process model for product strategy development supporting productization of software previously tailored to the needs of specific customers. The process model was developed with the commissioner company in mind, and only demonstrated at the commissioner company, where there was some initial understanding of the market for the product, and the functionality that it may offer in the SaaS version. Chapters 6 and 7 discuss the limitations of the model outside of the productization context and suggest modifications to expand its scope to brand new product development.

The thesis suggests certain tools to structure product strategy discussions and design. The thesis does not provide a taxonomy of all product management tools available. The goal is to find tools that work and demonstrate how the tools can be applied. The thesis does not claim that each tool is the best possible tool for its purpose. The thesis only claims that the selected tools are effective for their purpose. Alternative tools are mentioned where appropriate, but no focus has been placed on conducting a comprehensive taxonomy and comparison of product management tools. The development of a comprehensive taxonomy is suggested as a possible direction for future research.

1.3. Method

The research method employed in this thesis is design science research. Design science is defined by Johannesson and Perjons (2021) as “the scientific study and creation of artefacts as they are developed and used by people with the goal of solving practical problems of general interest.” The desired outcome is not only a novel artefact itself, but also the knowledge about the artefact, and the effects the artefact has on its environment (Johannesson and Perjons, 2021). Johannesson and Perjons (2021) define a specific design research process which contains 5 steps:

1. Explicate Problem
2. Define Requirements
3. Design and Develop Artefact
4. Demonstrate Artefact
5. Evaluate Artefact

The research process is introduced in more detail in Chapter 3.

1.4. Structure

The thesis is structured as follows. Chapter 2 provides the theoretical background for the rest of the work. The concepts related to product strategy, the ISPMA framework, SaaS and B2B software, and tools available for product strategy design are discussed. Chapter 3 describes the research method in general and discusses the first two steps of the design research process

by Johannesson and Perjons (2021). The research problem is explicated, and requirements for the desired artefact are listed. Chapter 4 presents the proposed process model for initial product strategy development. Each phase of the process model is explained. Chapter 5 demonstrates the process model by applying it to the commissioner company case. Chapter 6 evaluates how well the proposed model fulfilled the stated artefact requirements and achieved the research goal. In Chapter 7 the results of the study are discussed, and the answers to the research questions are provided. Chapter 8 concludes the thesis.

2. Background

This chapter introduces the concepts related to software process improvement, B2B and SaaS software products, productization, the SPMBOK, the product strategy, and tools available for strategy development. The knowledge collected in this chapter is later used during artefact design.

2.1. Software process improvement

Sommerville (2011, p.706) defines process improvement as “understanding existing processes and changing these processes to increase product quality and/or reduce costs and development time”. He writes that software development companies are under pressure to deliver quality products under tight schedules and within reasonable budgets. Companies look for process improvement approaches to accelerate product development, improve quality, and reduce costs (Sommerville, 2011, p.706). Ebert and Brinkkemper (2014) argue that software companies focus too much on project execution, technologies, and features, and neglect sufficient understanding of markets, value, and products. Ebert and Brinkkemper (2014) and Sommerville (2011, pp. 721-728) describe well-established methods for improving project execution. Lean and agile approaches provide cost-effective process improvement techniques for small and medium-sized organizations. For larger enterprises, the complex and maturity-focused Capability Maturity Model Integration (CMMI) model provides guidelines for process improvement (Ebert and Brinkkemper, 2014; Sommerville, 2011, pp.721-728). Ebert and Brinkkemper (2014) argue that using these techniques, a company may deliver a project to the customer or market on time and within budget. Without proper value and market understanding, however, such products may not sell as well as expected, or dissatisfy the customers (Ebert and Brinkkemper, 2014).

According to Paajoki (2020), process improvement of product management received less focus in academic research compared to project execution until recently. Maturity matrices and competence models have been developed to gauge the maturity of product management processes within companies (Paajoki, 2020). Ebert and Brinkkemper (2014) observed that the implementation of the product management role in software organizations is conducted in three phases: agreeing on foundations, preparation and piloting, and roll-out. In the

“foundations agreement” phase, stakeholder buy-in is acquired. The new product management role is defined in relation to other functional units of the company and communicated across the organization. During preparation and piloting, new training materials, self-assessment tools, templates and success stories for reference are added to the established product lifecycle. Finally, during roll-out, the product managers take on their duties in accordance with the SPM framework (Ebert and Brinkkemper, 2014).

An industry case study monitored 178 software projects over three years and found that organizations in the roll-out phase outperform the companies in “foundations agreement” phase by 85% in average delivery delay, 36% in average project duration, and 82% in quality of the shipped software products (Ebert and Brinkkemper, 2014). This result indicates that establishing a product manager role and integrating product management into the established product lifecycle helps companies build products faster and achieve better quality. Paajoki (2020) surveyed and interviewed software professionals in organizations that implemented the SPM framework. The study revealed that the framework adoption brought multiple benefits, including improvements in communication, customer satisfaction, employee satisfaction, increased customer value, and reduced number of product defects.

Ebert and Brinkkemper (2014) identified several success factors for the implementation of the product manager role. One of those factors is the updated standardized product lifecycle having clear interfaces, milestones, and governance. The implementation of product management practices must be supported by examples, templates, and training which is tailored to the business model and market positioning of the company (Ebert and Brinkkemper, 2014, p.17).

2.2. Productization of customer-specific software

Artz et al. (2010) define productization as the transformation of software tailored to the needs of specific customers into a standard software product. The transformation is driven by the recognition of similar needs and wishes of multiple customers (Artz et al., 2010). The researchers proposed a 6-stage productization process presented in Figure 1.

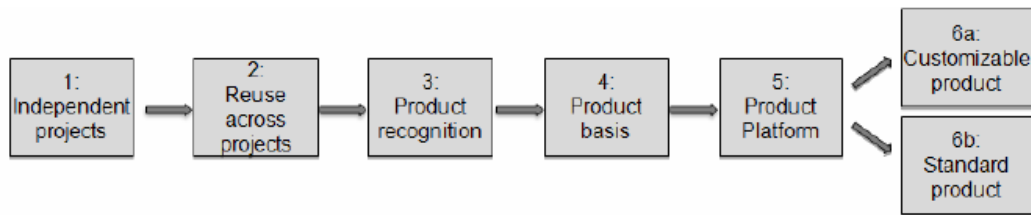


Figure 1. Productization process (Artz et al., 2010)

Below, a brief definition by Artz et al. (2010) is provided for each phase. The thesis will refer to these definitions when describing the software system undergoing productization in Chapter 4.

1. Stage 1 - Independent projects. At this stage, the company is working on independent projects that have no standard or reused features. Each project follows its own schedule and may have a separate technology stack. Each product follows its own customer requirements and undergoes customer acceptance at the end (Artz et al., 2010).
2. Stage 2 – Reuse across projects. Companies begin to reuse functions, features, and modules of past projects. This speeds up new project development and improves quality, assuming the reused elements were well-made and tested during previous projects. At this stage, the number of customized elements is still significantly larger than that of the reused ones (Artz et al., 2010).
3. Stage 3 – Product recognition. The number of standardized elements becomes greater than the customized parts. A company notices overlap in requirements of different customers and forms an initial understanding of a standard product. This is the first stage of product development where certain market boundaries and product characteristics are determined (Artz et al., 2010).
4. Stage 4 – Product platform. A product platform is defined in this context as “a set of features that form a common structure, from which a stream of derivative products can be efficiently customized, developed and produced (Artz et al., 2010).” The share of standard software elements increases. The company still works on customer-specific projects, and the number of customized features is still high, but the product is recognized, and the main core of the product is developed. The company starts to

gather market requirements to eventually prioritize market needs over the needs of specific customers (Artz et al., 2010).

5. Stage 5 – Standardizing product platform. The company begins to shift toward market orientation and releasing a standard product to the market. More features are added to the standard product core. The concept of product releases is introduced at this stage (Artz et al., 2010).
6. Stage 6a – Customizable software product. This is one of the two possible final stages of productization. The product is released to the market, and each customer using the product gets the same release of the software. But the product requires some customization, integration or installation and is difficult to sell “off-the-shelf”. Some customization layer is added on top of the standard product (Artz et al., 2010).
7. Stage 6b – Standard software product. The second possible final stage of productization. The product is standardized based on market needs. While the product is not customizable, it may provide certain configurable parameters to customers. The term “tailorability” will be introduced in Section 2.5.2 as an umbrella term for customizability, configurability, and composability (Artz et al., 2010).

The description of the 6 stages can be used by companies to estimate their position in the productization process.

2.3. Business-to-business Software-as-a-Service

Buxmann et al. (2013) describe the software-as-a-service model as “providing a standard software solution to customers as a service over the Internet”. The concept of delivering applications, storage and computing power over a network has first been introduced in the 1960s. Implementations by MIT, IBM and General Electric did not receive wide adoption at the time. The on-premise model became dominant with the emergence of personal computers and private servers in the late 1970s (Saltan, 2021). In the 1990s, application service providing (ASP) aimed to achieve a similar goal of delivering services over a network. Nowadays, the advancement of service-oriented architectures and open standards and the ubiquity of the Internet made the SaaS model widely adopted (Buxmann, 2013; Saltan, 2021).

Saltan (2021) analysed multiple existing definitions of SaaS, provided by the United States National Institute of Standards and Technology (NIST) and Gartner, as well as SaaS providers Cisco, HubSpot, Salesforce and Microsoft, and summarized the key characteristics of the SaaS model.

1. Multitenancy – a mode of operation of software where independent instances of one or more applications run on a shared platform. The instances are physically integrated, but logically isolated. SaaS customers share the IT infrastructure provided by the SaaS vendor. The customers share resources on network, host, and application levels, but their data is separated (Gartner, 2012; Saltan, 2021).
2. On-demand measured self-service – SaaS applications and resources are allocated to customers automatically, on demand. The resources are controlled, monitored, and optimized by the SaaS provider (Saltan, 2021).
3. Broad network access – required services are available over the Internet whenever the customer needs them, and on different types of devices, such as personal computers and smartphones (Saltan, 2021).
4. Elasticity and scalability – SaaS solutions can scale up and release resources to meet customer demand. Virtualization technologies add a layer of abstraction between applications and the physical resources of the SaaS provider. This allows the provider to use their resources efficiently and ensures customers have access to resources when they need them (Buxmann, 2013; Saltan, 2021).
5. Subscription-based pricing – SaaS customers pay subscription fees at regular intervals in exchange for the right to use the software. User and usage-based metrics are employed to determine the recurring price. The subscription price typically covers all associated services, such as maintenance, customer support, and data storage (Saltan, 2021).

The term business-to-business is used to describe arrangements between different businesses, as opposed to between businesses and the general public (Cambridge Dictionary, 2022). Saltan (2021) notes that the term SaaS is rarely used outside of the B2B context when describing software. In the B2C area, the terms “cloud services” and “online services” are often used instead (Saltan, 2021).

2.4. Product strategy

The Cambridge Dictionary (2019) defines a strategy as “a detailed plan for achieving success in situations such as war, politics, business, industry, or sport, or the skill of planning for such situations”. A second definition is “a way of doing something or dealing with something”. Indeed, a software product strategy is a high-level plan that lets companies achieve the vision for their products (Pichler, 2016). Kittlaus and Fricker (2017) write that a product vision describes what the product will be at the end of a certain strategic timeframe. The vision describes the value the product will bring to the customers and the vendor in a succinct, inspirational way. The purpose of designing the product strategy is then to determine the path to achieving this vision (Kittlaus and Fricker, 2017). The fast pace of technological innovation forces product managers to make long-lasting and financially impactful decisions about their products in the face of relative uncertainty. Having a clear strategy for several years into the future provides a basis for making those decisions, and aligns the stakeholders involved in product development (Kittlaus and Fricker, 2017).



Figure 2. Elements of a Product Strategy (Pichler, 2016)

Figure 2 positions the strategy at the centre of three main components. Those components are the business goals of the software vendor, the key features of the product that make it stand out, and the market for the product, with its specific needs. The business goals justify the existence of the product to the company, and having a clear goal helps select the appropriate KPIs to measure product performance (Pichler, 2016).

Kittlaus and Fricker (2017) define a product as “a combination of goods and services, which a supplier or development organization combines in support of its commercial interests to transfer defined rights to a customer.” While laborious, this definition highlights two significant aspects. The phrase “in support of its commercial interests” is used to emphasize that the product itself does not necessarily require a payment, but might instead increase market penetration, help sell other products, or increase overall brand equity (Kittlaus & Fricker, 2017; Pichler, 2016). One example is the Google Chrome browser, which is offered for free, but reduces Google’s dependency on third party browsers and gives company more control of how people access the Internet (Pichler, 2016, p.18). The phrase “to transfer the defined rights to the customer” highlights the fact that the software vendor maintains the ownership of the product, while the customer obtains the right to use it under a given service license agreement (Kittlaus and Fricker, 2017).

A software product manager is responsible for the management of software products to achieve sustainable economic success over the entire product lifecycle (Kittlaus and Fricker, 2017). The following Section will describe the 8 knowledge areas related to the product strategy that a product manager must be aware of.

2.5. The ISMPA SPM framework and the SPMBOK

A Body of Knowledge is a structured and comprehensively defined set of knowledge about a certain discipline and associated professional areas. Bodies of Knowledge provide guides to concepts, terms, competencies, and activities within their domain (Saltan and Smolander, 2019). The Software Product Management Body of Knowledge (SPMBOK) has been developed over the recent years, based on industry best practices and academic research. SPMBOK is continuously improved as a result of implementing, optimizing, and monitoring product management activities at companies worldwide (Ebert and Brinkkemper, 2014).

Multiple frameworks exist to describe the responsibilities of a software product manager. The SPM framework by ISPMA consolidates multiple preceding frameworks and provides a holistic perspective on the product manager role (Paajoki, 2020). Ebert and Brinkkemper (2014) describe the framework as the “underlying knowledge area framework of the SPMBOK”. The framework is presented in Figure 3. Each field in the table is a knowledge area in SPM (Ebert and Brinkkemper, 2014). Each column of the framework corresponds to a functional area of a software organization. Vertically the responsibilities in each column

move from more long-term and strategic at the top to day-to-day operational short-term at the bottom (Kittlaus and Fricker, 2017).

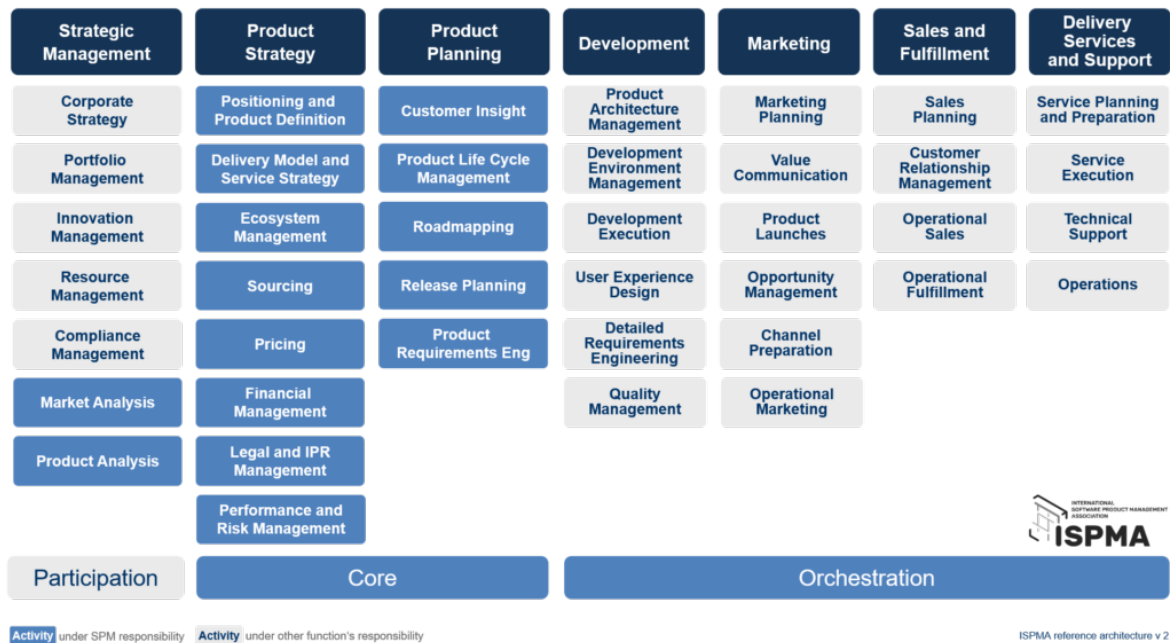


Figure 3. ISPMA SPM Framework V.2.0 (ISPMA, 2021a)

The rest of the section provides definitions of the key concepts related to each of the 8 product strategy knowledge areas outlined in the framework. The definitions are largely based on the official 2021 ISPMA syllabus, used to conduct training for aspiring product management experts. The chapter also refers to the ISPMA-compatible study guide and handbook by Hans-Bernd Kittlaus and Samuel Fricker, the first edition of which was published in 2017. Hans-Bernd Kittlaus has also kindly provided excerpts from the second edition of the handbook before its publication. In the final weeks of the thesis work, the full second edition of the handbook was published. Looking at the three primary sources for the chapter — the 2017 first edition of the handbook, the 2021 edition of the syllabus, and the 2022 second edition of the handbook — it is possible to see how the SPM framework evolved, with elements getting renamed, the scope of different elements readjusted, and focus of certain areas shifted or expanded. Notable changes in the newer syllabus and handbook include the expanded focus on software architecture as part of product strategy discussion, as will be explained in the “positioning and product definition” section below.

2.5.1. Positioning and product definition

A dictionary definition is once again helpful. Positioning is “the way that customers think about, or the way that a company wants customers to think about, a product in relation to similar products or to competitors' products (Cambridge Dictionary, 2019a)”. Kittlaus and Fricker (2017) write that fundamentally positioning is concerned with the value proposition of a product. The value proposition needs to be formulated for each identified market segment, and the sales channels for each of the segments need to be established. In managing the positioning of a product, a product manager needs to work together with Marketing, Sales, and executive management (Kittlaus and Fricker, 2017).

Other sources compare positioning to an elevator pitch for the product, a succinct statement that “frames the product within a known category and against a competitive landscape (Bavaro and McDowell, 2021)”. The purpose is to communicate to the customer what problem the product will solve for them, and to “strongly resonate” with each customer segment. “Yes! I do have that problem and I do want that solution!” is the expected outcome (Bavaro and McDowell, 2021).

Positioning must remain consistent across the marketing materials for the product and the talking points utilized by the sales department. It is the role of a product manager to ensure that positioning is clear to all stakeholders throughout the organization (Bavaro and McDowell, 2021). Kittlaus and Fricker (2017) mention that positioning is also relevant within a company's product portfolio, especially if the functionality of several products overlaps to a certain extent.

According to the latest edition of the ISPMA syllabus (2021b), a product definition elaborates on the product vision but does not yet become a complete product specification. It encapsulates the following concepts:

- Functional scope – the features included in the product. Defines which features are in and out of scope for the product. As with other parts of a strategy, the functional scope can be revised over the product's lifecycle, with features added or removed (Kittlaus and Fricker, 2017).

- Quality scope – the intended level of quality for the product. Often given as a comparison to the existing competition. “At least as good” or “5% faster than the previous version” are acceptable estimations (Kittlaus and Fricker, 2017).
- Intended use and users – a list of possible users and their characteristics and preferred use cases. These can be developed into more developed personas at later stages of UX design (Kittlaus and Fricker, 2017). An example is “for teachers and students to submit and grade homework”, if the product in question is in the domain of education.
- User experience (UX) design scope is defined similarly to the quality scope, with the focus on the quality of the user experience. Again, a comparison with the competition or other products in a product line is appropriate, e.g., “needs to be as good as our other product X” or “the reported user satisfaction must be higher by 10% compared to product X (Kittlaus and Fricker, 2017)”.
- Offering architecture describes each separately priced component of the whole product offering and the tailorability options in line with the tailorability strategy (Kittlaus, 2022).
- Business architecture closely overlaps with software architecture and can involve the logical data model, the process model, and the business object model for the software. It is developed collaboratively by a product manager and a technical architect, with the product manager taking the lead and the technical architect ensuring feasibility (Kittlaus, 2022). The required contents of the business architecture are described only briefly in ISPMA literature (see ISPMA, 2021; Kittlaus, 2022). Kittlaus (2022, p.72) states that the business architecture is domain-specific, and “covers logical data model, process model, business object model etc.” (sic). The use of the term “domain-specific” may imply that the documentation of this section may vary significantly based on organization and industry.

In the 2017 edition of the Kittlaus and Fricker handbook, the offering and business architecture aspects are largely absent from the product definition description.

Two more aspects can be described as part of the product definition – the technical constraints and the compatibility scope (Kittlaus and Fricker, 2017). Technical constraints

describe the technical decisions that are made early on and impose restrictions on product development. Compatibility scope describes the function, data, and interface compatibility between older and newer versions of the product. A vendor can choose to maintain upwards compatibility, where the data, functions, and interfaces of the older version of a software product are compatible with newer versions of the product. For example, documents created via an older version of a word processing tool can still be opened and edited in a newer version of the tool. Without upwards compatibility, the vendor should instruct the customer on how to carry out the migration to a newer version (Kittlaus and Fricker, 2017). The downwards compatibility is the opposite, where the data, interfaces and functions of newer versions are compatible with the older versions. While upwards compatibility is typically expected, downwards compatibility is more rarely so (Kittlaus and Fricker, 2017).

Kittlaus (2022) writes that product definition is an iterative process, and the result evolves over the strategic timeframe. The definition becomes more stable for products at later stages of their lifecycle, but for new products it is frequently changed during a dynamic learning process. Product definition of new products is focused on an iterative creation of a Minimum Viable Product (MVP) – a minimum feature set that at least some customers are willing to buy in its first release (Kittlaus, 2022).

2.5.2. Delivery model and service strategy

The delivery model is “a description of the mechanisms in which a product is made available to customers (Kittlaus and Fricker, 2017)”. The ISPMA syllabus (ISPMA, 2021) states that the two options for this are a licensed product or a service offering, such as SaaS. A hybrid of both options is also possible. The delivery model should also cover the tailorability architecture of the software product (ISPMA, 2021b).

According to Kittlaus and Fricker (2017), the service strategy covers the human services related to the product. These provide additional value to the customer by ensuring the software product can be taken into use quickly and operate with as little friction as possible from the customers’ point of view. The service strategy may involve customer support, training seminars, installation, customization, and maintenance. These services can be priced separately from the core product. Maintenance of SaaS products is not charged separately but is included in the recurring charges (Kittlaus and Fricker, 2017).

ISPMA (2021b) describes tailorability as “the enablement of the product for customer- or market-specific adaptations by providing properties that can be changed after system development” and draws a distinction between 3 categories of tailorability:

- Configuration – tailoring the product by adjusting its parameters
- Composition – tailoring the product by adding or arranging components
- Customization – adding or changing program or descriptive code

Tailorability architecture is present in the 2017 edition of the Kittlaus and Fricker handbook under a different name – the customization scope.

2.5.3. Ecosystem management

Ecosystem management positions the product within existing software ecosystems. Most companies are niche players that do not aim to compete with ecosystem keystones and dominators such as Google or Apple, but rather specialize in their narrow area of expertise. Part of the development of a product strategy is determining possible partners in the ecosystem (Kittlaus and Fricker, 2017).

2.5.4. Sourcing

According to the ISPMA syllabus (2021b), sourcing addresses the provisioning of resources required to develop the product. These can be procured in-house or outsourced. This applies to human resources and software components. Outsourcing human services is justifiable to acquire expertise unavailable in-house, increase the working capacity, and save costs. Acquiring external software components can reduce time to market and increase quality when lacking the time or expertise to develop certain elements of the product internally (ISPMA, 2021b).

2.5.5. Pricing

Saltan (2021) defines pricing as “the process of decision-making in determining the monetary compensation and related conditions of the goods and services the customer is offered”. Multiple classifications exist to systematize pricing strategies. A common differentiation is between value-based, market-based, and cost-based pricing (Saltan, 2021).

- Value-based pricing means the software is priced according to the value it brings to the customers. The value perceived by the customer can be subjective, allowing for charging different customers different rates (Saltan, 2021). Kittlaus and Fricker (2017) write that for software products value-based pricing is the only approach. High fixed costs and comparatively low variable costs are unique traits of software products. An initial investment into product development is compensated by the high profit margins when the product is released to the market (Kittlaus and Fricker, 2017). Value-based pricing is also recommended for SaaS products in consultancy publications by Accion (2015) and PWC (2013).
- Market-based pricing means that the product is priced according to the supply and demand status of the market. Prices offered by the competition and the price sensitivity of potential customers are considered. In literature, market-based pricing is sometimes differentiated from competition-based pricing (see Saltan and Smolander, 2021). If there is no competition in the target market, the willingness to pay of the potential customers is considered as the basis for setting the price (Saltan and Smolander, 2021; Saltan, 2021).
- Cost-based pricing means the software is priced according to the costs incurred by the company when developing the software, plus the profit margin. While easy to calculate and set, this kind of pricing applies to professional services but not software products. This is because once the software development effort is complete, there is little variable cost. Using this low variable cost as the basis for cost-based price estimation leaves a lot of potential revenue “on the table” for the software vendor (Kittlaus and Fricker, 2017; Saltan and Smolander, 2021).

The strategic pricing pyramid presented in Figure 4 describes different levels of value-based pricing. At the basis of the pyramid is value creation, where customer segments and value delivered to those segments are clearly defined. The price structure describes how the price may vary for different customer segments, or according to different value metrics. Here many approaches exist, including pay-as-you-go (usage-based), feature-based, tiered, one-time or periodic, or even free (Kittlaus and Fricker, 2017; Saltan and Smolander, 2021). Price and value communication ensure that the price is understood, acceptable to and accepted by the target market. The pricing policy sets governance criteria for the price. This can clarify the conditions for discounting and put measures in place to prevent abuse of these

conditions. Finally, the price level establishes the price for each customer according to the price structure (Kittlaus and Fricker, 2017).

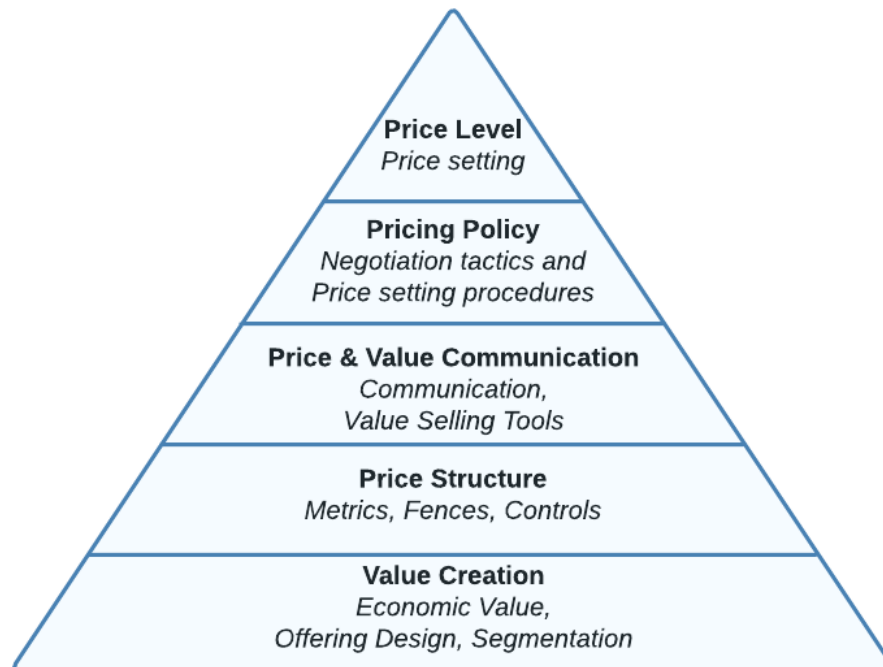


Figure 4. The strategic pricing pyramid (ISPMA, 2021b)

According to Saltan and Smolander (2021), “the implementation of proper SaaS pricing is one of the most complex activities that any company can attempt”. Small and medium-sized companies lack the required resources and proper understanding of pricing concepts to make pricing strategy decisions. The pricing process is haphazard and undermanaged as a result. Moreover, the academic literature does not provide comprehensible and proven methods to select the right pricing approach from the myriad of available options (Saltan and Smolander, 2021).

2.5.6. Financial management

The financial management element of product strategy was added to version 2.0 of the ISPMA SPM framework. In older versions of the framework, this element was referred to as “Business case and costing”. The new financial management element encapsulates a wide variety of aspects that need to be considered to ensure the long-term economic success of a software product. A product manager must control the balance between investments into product development and profits generated by the product. Both the short-term and the long-

term perspectives need to be considered (ISPMA, 2021b). The ISPMA syllabus (2021b) mentions four aspects of financial management – business model, business case, business plan and cost management. Additionally, the syllabus mentions techniques that can be used to justify investment decisions.

The business model describes how an organization creates, delivers, and captures value. A business model can be considered on different levels: for a company as a whole and for specific products created by the company. Documenting a business model for one of the products makes sense especially when the new product requires a company to adopt a business model that hasn't been used by the company previously (ISPMA, 2021b).

According to Pichler (2016), while the business model specifies the cost factors and revenue streams, it does not specify the actual numbers of expected costs and revenues. These numbers are projected within a business case. The business case becomes challenging to describe accurately when products in question are disruptive, operating in uncharted waters where potential costs and revenues are difficult to predict. For such products, Pichler recommends justifying the investment by producing a sound business model and describing the inaction risk – the benefits that the company will not reap if the product is not created (Pichler, 2016).

2.5.7. Legal and IPR management

While a product manager is typically not a legal expert, they need to be aware of the main legal risks that can harm the long-term success of software products. Within the scope of product management, the goal is to determine the most important questions that a product manager can discuss together with dedicated legal experts (Kittlaus and Fricker, 2017).

The core legal areas requiring attention are described by Kittlaus and Fricker (2017) as follows:

- **Contracts.** Contracts are made between a software vendor and its customers. The type of contract is based on the delivery model, which can be SaaS or a licensed software product. Licensed products require license contracts, while for SaaS a service contract is needed instead. The service contract may be negotiated with a customer or offered in the form of “standard terms and conditions” to a mass market. A SaaS contract typically contains a service level agreement (SLA), which defines what

services the customer may expect from the provider. An SLA may contain the definition of the functional scope, the availability commitment, the maximum possible length of downtime, the level of quality and performance, the speed and scope of customer support, backup frequency and disaster recovery plans, as well as the possibility of requesting new or changed functionality, and the estimated times for such implementations or changes. SLAs are beneficial to SaaS customers if they describe the compensation that the vendor must pay the customers when they fail to meet the terms of the agreement – the liability. The types of charges may also be specified in the service contract. This may not specify the exact price, but the price metrics according to the pricing policy (Kittlaus and Fricker, 2017).

- Protection of intellectual property. A software vendor has multiple ways available to protect its intellectual property. The source code itself is protected by copyright, prohibiting reusing it unchanged. But the algorithms, functions and ideas are not protected by copyright. To protect ideas and technical concepts, a patent can be obtained. Trademarks can be acquired for brand names. Companies can prevent employees from sharing internal knowledge through non-disclosure agreements (Kittlaus, 2022).
- Open source. Using open source software is a widespread way to reduce development costs and time. But a software vendor must be careful when selecting open source components. Each open source component may specify license terms, which impose restrictions on distribution. Certain licenses prohibit unlimited commercial use, or require the entire source code of a project using an open source component to be made publicly available even when only a few lines of the open source component code are used. A product manager must be aware of different license types and consult the legal department for absolute clarity. A list of approved licenses or approved open source components may be developed by the legal department of a software vendor (Kittlaus, 2022).
- Data protection. A SaaS software vendor must comply with the regional legislation on collecting, processing, and using the personal data of the users of the software. In Europe, the GDPR (General Data Protection Regulation) legislation sets the rules in this domain. GDPR was the most extensive data protection legislation at the time of its coming to effect in 2018, and other countries have since developed their own

legislation using GDPR as the model (Kittlaus, 2022; Wolford, 2018). GDPR defines personal data as “any information that relates to an individual who can be directly or indirectly identified (Wolford, 2018)”. For a software vendor, the GDPR terms “data controller” and “data processor” are significant. A data controller determines the purposes for and the means of processing the data (the why and the how). A data processor processes personal data on behalf of the controller (European Commission, 2019; Wolford, 2018).

When a customer organization uses SaaS, personal data is processed on the server infrastructure of the service provider. As a rule, in this case, the customer organization is the data controller, and the SaaS provider is the commissioned data processor. GDPR mandates that a contract is made between the controller and the processor. In the contract, the SaaS provider commits to transparency, compliance with the directions of the customer organization, and enforcing appropriate security measures (European Commission, 2019; Kittlaus, 2022). The contract must also describe what happens to the data when the contract is terminated (European Commission, 2019). The terms of transferring data outside the European Economic Area (EEA) and the usage of cookies are covered by GDPR too and must be addressed by the product manager together with the legal experts of their organization (Kittlaus, 2022; Koch, 2019).

Several other risks are mentioned as increasingly relevant. Those are governance, supply-chain, delivery commitments, laws on general terms and conditions, and blacklisting of countries for specific software components (Kittlaus and Fricker, 2017).

2.5.8. Performance and risk management

Performance means continuous analysis and tracking of the business performance of the product after the strategy is defined. For this purpose, key performance indicators (KPIs) are selected (Kittlaus, 2022). Pichler (2016, p.42) compares operating a software product without KPIs to “driving with your vision blurred”. Kaplan and Norton (1996) draw another traffic comparison and compare managers to pilots. Pilots must process information from a vast set of indicators to control airplanes. The same applies to managers, who need to form a holistic perspective on how the product is performing.

Product management literature often recommends using a Balanced Scorecard (BSC) to track product performance (see, e.g., Pichler (2016) or Kittlaus (2022)). Both Kittlaus and Pichler provide their own modifications of the original BSC introduced by Norton and Kaplan in the 1990s. BSC calls for supplementing financial metrics of product and business unit performance with other dimensions for a holistic perspective. Parmenter (2015) traces this balanced performance management approach to an even earlier methodology Hoshin Kanri. Hoshin Kanri was developed by a multinational Japanese corporation, and also provides four perspectives, mirrored in the BSC. Table 1 shows the variations of the BSC dimensions found in management literature.

Table 1. Variations of performance metric categories in management literature

Hoshin Kanri (before 1996)	Norton and Kaplan (1996)	Parmenter (2015)	Pichler (2016)	Kittlaus (2022)
Cost Objective and Measures	Financial	Financial Results	Financial	Financial
Quality Objectives and Measures	Customer	Customer Focus	Customer	Customer / Market
Delivery Objectives and Measures	Internal Business Process	Internal Process	Product and Process	Internal Business / Organizational
Education Objective and Measures	Learning and Growth	Innovation and learning	People	Innovation and learning
		Staff Satisfaction		
		Environment and Community		

Kittlaus (2022, p.129) writes that the BSC is a comprehensive framework, and it is complex and time-consuming to define all the relevant indicators and the ways to measure them. The

measures should be sufficiently easy to measure and provide decision support (Kittlaus, 2022, p.131). While BSC is a prominent approach to performance management, it is far from the only one. For example, Doshi (2021) recommends monitoring six categories of metrics different from those offered in a BSC: health, usage, adoption, satisfaction, ecosystem, and outcome. PuMP (Performance Measurement Process) by Stacey Barr and the “Winning KPIs methodology” by Parmenter argue that most organizations follow an ad-hoc approach when selecting performance measures, underestimating the effort and rigour that is required to select and monitor proper metrics (Parmenter, 2015). Parmenter (2015) calls out the multiple shortcomings of the BSC and the ways in which it is misunderstood. He suggests two additional angles (Environment and Community, and Staff Satisfaction) and argues that many metrics do not fit into a single perspective. He also advocates establishing a KPI team and a chief measurement officer in charge of performance management. Still, Parmenter admits that adopting the BSC is “certainly a step in the right direction (Parmenter, 2015, p.63)”. Pichler also incorporates staff satisfaction into his variation of the BSC by readjusting the “Learning and Growth” perspective into “People”.

Regardless of the exact grouping of KPI categories, literature typically provides lists of example KPIs for each category, with the addendum that the list is not comprehensive and KPIs vary based on the specific context and business goals (see, e.g., Doshi, 2021; Kittlaus, 2022; Parmenter, 2015; Pichler, 2016). Appendix 1 lists example KPIs as described by Pichler (2016). At a minimum, even when following the ad hoc approach for KPI selection, several best practices can be followed:

1. The KPIs should be tied to business goals. The business goals are the reason the product is being created, and ensuring that the product is meeting these goals should be a top priority (Pichler, 2016).
2. The KPIs should be measurable. To effectively evaluate performance, it is necessary to understand how well the product is reaching the stated business goals. It should be possible to tell how well the product is reaching certain targets. Here, a distinction needs to be made between qualitative and quantitative KPIs. As the terms imply, quantitative KPIs deal with numbers, such as revenue or the number of daily users. Qualitative KPIs can come in the form of user reviews or feedback given to customer service (Pichler, 2016). Setting specific numerical targets can be difficult, especially for new products. Using ratios and ranges can help in such cases (Croll and

Yoskovitz, cited by Pichler, 2016). For example, a revenue target could be formulated as “the product shall increase overall company revenue by 5 to 10% within one year after launch (Pichler, 2016)”. Doshi (2021) cautions against trying to precisely predict numbers and suggests setting qualitative goals that are “the closest possible proxy to the numbers you want to achieve”.

3. Vanity metrics should be avoided. Vanity metrics are the ones that make the company look good, but don’t provide value or actionable insight (Kittlaus, 2022; Pichler, 2016). For example, a large number of app downloads may seem good and prove marketing efforts successful. But it fails to reveal cases when users lose interest and stop using the app after giving it a try for a few days or even minutes (Pichler, 2016).
4. The KPIs should be loosely grouped into several categories to ensure a holistic perspective. The holistic approach reduces the risk of overlooking certain important aspects (Doshi, 2021; Pichler, 2016).

Risk management refers to tracking risks associated with the product, and swiftly reacting when necessary (Kittlaus, 2022). Kittlaus (2022) identifies the three types of risks as product, customer, and market risks. Depending on the lifecycle stage of the product, different types need to be prioritized. For new product development, the main risks relate to finding the problem-solution fit and offering the right value proposition to the selected customer segments. A product manager is responsible for identifying and mitigating these risks and making contingency plans. This needs to be done routinely and periodically (Kittlaus, 2022). Sommerville (2011) proposes a risk management process presented in Figure 5. This process was designed for software project management, but it is sufficiently generic to be applicable in other disciplines, including software product management.

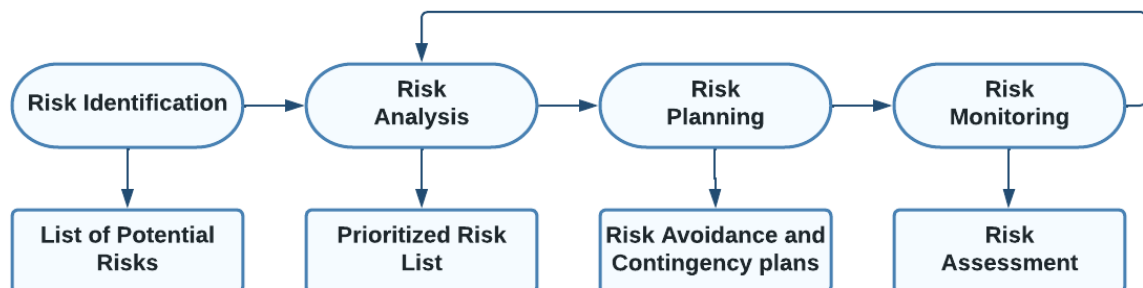


Figure 5. Risk management process (Sommerville, 2011)

Pichler (2016) posits that new product success is the intersection of the right technology and features, the right market and needs, and the right business goal. Each of these categories has associated risks, which can be identified and prioritized. The sample questions that can highlight those risks are presented in Table 2.

Table 2. New product risks and sample questions (Pichler, 2016)

Risks	Sample Questions
Market and Needs	<ul style="list-style-type: none"> • Will removing the problem you want to solve make a real difference to the customers and users? • Is the benefit the product creates something people would not want to miss once they have experienced it? • Are you confident that you have selected the right market segment, and that you are addressing the right people? • Is the target group clear-cut? • Are you able to tell who is in your target group and who is not? • Do you have a rough idea of how big the market is? • Are there any major market entry barriers?
Features and Technologies	<ul style="list-style-type: none"> • Will the product be able to do a great job for the users and customers? • Can you list the top three features that will make people want to use or buy the product? • Does the product offer a clear and compelling advantage over the competitors' products? • Are you clear on the branding strategy? • What are the key characteristics of the desired user experience? • Is it feasible to develop the product? • Are the technologies required available? • Are they mature enough? • Does your organization have the necessary skills to use them effectively? • Are enough people with the right skills available, and if not, can you recruit them?
Business Goals	<ul style="list-style-type: none"> • Are you confident that it's worthwhile to develop, market, sell, and support the product? • Are you clear on the business goals the product should deliver? • Can you quantify and measure the desired business benefits? • Do you know which one is your number one goal? • Do you understand how the goals can be achieved? • What business model will you use? • Are you confident that the business model will work?

Creating and testing prototypes, interviewing potential customers, and observing the usage of the software are valid risk mitigation techniques (Pichler, 2016). For risk identification, Pichler suggests the “Red Dot Game”. Stakeholders at the company are presented with the product strategy document and asked to place red dots next to the segments or statements in the strategy that they are least sure about. Each participant can only place three dots in total. By looking at the most highlighted areas it is possible to rate the riskiest parts of the strategy.

2.6. Documentation of the product strategy

Many approaches to documenting the strategy are mentioned in the literature. Kittlaus (2022) writes that product strategy should address the following elements:

1. Positioning and Product Definition
2. Delivery model and Tailorability
3. Service Strategy
4. Sourcing
5. Pricing
6. Financial Management
7. Ecosystem Management
8. Legal and IPR Management
9. Performance and Risk Management
10. Roadmap

According to Kittlaus (2022), elements 1–4 should be included in detail in a single cohesive strategy document. Elements 5–9 are typically only mentioned at an abstract level in that document. The precise descriptions of these aspects are kept separate (Kittlaus, 2022).

Bavaro and McDowell (2021) describe their approach to making product documentation at American “Big Tech” companies. They stress the importance of making the document easy to read and making it the focal point for feedback. Diagrams, bullet lists, and concise language should be used to ensure the document is well understood. They recommend gathering feedback on the document asynchronously, because that allows to collect all the

comments before preparing a response and making changes. But arranging a meeting to discuss the document is also possible, if necessary (Bavaro and McDowell, 2021). They recommend highlighting the key trade-offs and decisions made during strategy development. Pointing out the most controversial bits of the strategy helps to elicit additional feedback, confirm or change the decisions, and ensure alignment (Bavaro and McDowell, 2021).

Introduction	2
Positioning and Product Definition	2
Who is the customer?	2
What customer needs are we addressing?	2
Problem and position statement	2
Functional scope	6
Quality scope	6
Intended use and users	6
User experience (UX) design scope	7
Offering architecture	7
Business architecture	7
Domain model	7
Business process models	9
Key decisions and trade-offs, controversial bits	10
Delivery model and Tailorability	10
Service Strategy	11
Sourcing	12
Pricing	12
Financial Management	18
Business goals 🎯	18
Costs 💰	18
Revenue streams 📈	18
Ecosystem Management	18
Legal and IPR Management	19
Data protection 🗝️	19
Contracts 📄	20
Open source components 🍷	20
Protection of intellectual property 🧠	21
Performance and Risk Management	21
Roadmap	23

Figure 6. Example table of contents for the product strategy document

Other forms of product documentation include the Business Model Canvas (Kittlaus, 2022), the Product Vision Board (Pichler, 2016), the lean product canvas, and a Product Requirements Document (PRD) (Bavaro and McDowell, 2021). Regardless of the template used, Bavaro and McDowell (2021) stress removing unnecessary sections, striving for clarity, and treating the document as a tool to clarify thinking about the product, organize ideas, and plan ahead. An example table of contents of a product strategy document is presented in Figure 6.

2.7. Software product strategy in B2B and SaaS context

Product management publications describe several challenges faced by product managers when developing and maintaining strategies for B2B software products. According to Kittlaus and Fricker (2017), the overall initial strategy needs to be sufficiently developed to gain first customers. B2B customers that make significant investments into new software need to see the long-term strategy for the software to be confident in their choice (Kittlaus and Fricker, 2017). The product vision in the B2B area typically spans longer compared to B2C (Kittlaus and Fricker, 2017).

A major challenge described in several sources (Kittlaus and Fricker, 2017; Bavaro and McDowell, 2021; Product School, 2019a; Product School, 2019b) is that the software vendor primarily communicates with the buyer, who is not always the end user of the software. A positive holistic approach for a B2B product takes both buyers and end users into account. Pichler (2016) suggests developing buyer personas distinct from user personas, because the characteristics and goals of the two groups may differ significantly.

A product manager has to spend a lot of time with B2B customer representatives and make connections between customer needs and opportunities for the business of the software vendor (Bavaro and McDowell, 2021). Often the customer expects personal contact with sales representatives on demand, since they are paying a high price for the software product (Kittlaus and Fricker, 2017). But – according to Kittlaus and Fricker (2017) – there is risk in such close customer contact. While maintaining close contact with customer representatives is a valuable means to receive quick answers and feedback, there is a chance of tailoring a product too much to the needs of a particular customer. A product manager must always keep the overall target market in mind (Kittlaus and Fricker, 2017).

The tailorability element of product strategy also has challenges in the B2B context. Kittlaus and Fricker (2017) mention that in the B2B area the customization options for a product are typically more extensive. The higher degree of tailorability brings additional complexity to product development. To maximize the value for B2B customers, core products frequently require add-on products and integrations with third-party products (Kittlaus and Fricker, 2017).

Salameh and Bass (2019) write that in the B2B context the software vendor and the customer are in a continuous communication process. Product requirements are continuously refined in discussions between the two parties. The vendor must continuously clearly convey its organizational capabilities to the customers. Misunderstandings of requirements and the capabilities of the supplier pose a risk for B2B product implementations, even for relatively simple products (Salameh and Bass, 2019). In smaller companies that try to keep the product definition open to experimentation, a particular challenge may manifest itself when the sales department makes a contractual commitment to develop specific features, thus forcing constraints on the product scope (Product School, 2019a).

When selecting metrics to evaluate the performance of the product, a major focus in the B2B area is on customer retention and preventing churn (Bavaro and McDowell, 2021). The KPIs in B2B context are focused on revenue, including the number of active premium users, annual recurring revenue (ARR), and the number of closed deals (Bavaro and McDowell, 2021). Measuring engagement with specific features can also be a valuable source of information on what features to prioritize to retain customers (Bavaro and McDowell, 2021). But A/B testing features in B2B products may pose a challenge in cases of mission-critical systems running in production environments (Product School, 2019a). The critical role of such systems requires precise functionality and high availability. Experimenting with features may not be acceptable in such cases.

When it comes to pricing, it may be difficult for a B2B vendor to gauge the prices of the competition (Kittlaus and Fricker, 2017, p.96). Pricing information is often not available publicly and is only obtained by contacting a sales representative directly.

Finally, a software vendor may attract customers in the B2B context by promising to ensure legal compliance in the customer's business area (Kittlaus and Fricker, 2017).

A conflict can be observed. In the B2B market, customers expect the products they purchase to fulfil their specific requirements. But the SaaS vendor wants to offer a standard product to all its customers. Customizing the product for each customer makes it non-standard and turns the work of the software vendor into providing professional services to each customer instead of a standard product. The concept of tailorability introduced in Section 2.5.2 bridges this gap between the standard product functionality and the unique customer requirements. It allows the customer to adapt the standard product for their specific needs (ISPMA, 2020).

2.8. Selected tooling

In this chapter, some of the tools that can aid product managers in strategy design are presented.

2.8.1. Problem and position statement template

Kittlaus and Fricker (2017) suggest the template presented in Table 3 for developing the product vision, having used it at a number of software companies. The template focuses on the problem space and the solution space. First, the problem faced by potential customers is described, independent from the solution. The template describes the pain points faced by the customers, and what a successful solution would look like from their perspective. Next, the solution is described, including its functional scope, use cases and unique value proposition (Kittlaus and Fricker, 2017).


Table 3. Problem and position statement template (Kittlaus and Fricker, 2017)










Problem statement	
the problem of	
affects	
the impact of which is	
a successful solution	
Position statement	
for	
who	
the	
that	
unlike	
our product	

The template ensures that the information required to formulate the product vision is collected. Once the information is collected, the filled template can be rewritten to make the text more concise and natural, if necessary. The text can then be used in product strategy documentation and marketing materials (Kittlaus and Fricker, 2017).

2.8.2. Product Vision Board

The Business Model Canvas tool was introduced by Osterwalder and Pigneur in 2010 and has since become widely influential and adopted in business education and practice (Carton, 2021). Multiple variations of the tool have been created and used over the years (Kittlaus and Fricker, 2017, p. 50). These variations include the Lean Canvas and the Product Vision Board (the latter offered by Pichler, 2016). These tools allow describing the strategy and business model for an entire company or a specific product by filling several boxes, each representing a different area that needs to be considered for a holistic understanding of the business model.

THE PRODUCT VISION BOARD EXTENDED 

 VISION What is your motivation for creating the product? Which positive change should it bring about?			
 TARGET GROUP Which market or market segment does the product address? Who are the target customers and users?	 NEEDS Which problem does the product solve? What benefit does it provide?	 PRODUCT What product is it? What makes it stand out? Is it feasible to develop the product?	 BUSINESS GOALS How is the product going to benefit the company? What are the business goals?
 COMPETITORS Who are your main competitors? What are their strengths and weaknesses?	 REVENUE STREAMS How can you monetise your product and generate revenues?	 COST FACTORS What are the main cost factors to develop, market, sell, and service the product?	 CHANNELS How will you market and sell your product? Do the channels exist today?


www.romanpichler.com This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License 
 Template version 09/15

Figure 7. The Extended Product Vision Board (Pichler, 2022)

The Extended Product Vision Board is presented in Figure 7. The vision is specified at the top. The next row defines the target market, the needs of that market, the product addressing those needs, and the business goals motivating the company to develop the product. The bottom row describes the competitors, revenue streams, cost factors, and channels to market and sell the product. Pichler also offers a simplified version of the board without the bottom row. In this thesis, the full version of the board is used to describe the product strategy in more detail – hence the “Extended” name in the title of the tool.

2.8.3. Strategy Canvas and ERRC grid from the Blue Ocean family of tools

A common way to look at an industry is competition-based. Michael Porter famously defined the 5 forces that can be used to analyse an industry. The competitive strategy described by Porter relates to the perspective on economics where long-term competition and imitation are the dominant forces (Burke, Van Stel and Thurik, 2016). Kim and Mauborgne (2005) challenge that perspective and argue that companies can discover new markets where they can grow their revenues without competition. Such markets are described as the desirable “blue oceans”, and the way of thinking overall is referred to as the “blue ocean strategy” (Burke et al. 2016; Kim and Mauborgne, 2005). Kim and Mauborgne (2005) claim that tools exist to create blue oceans systematically. They offer a set of frameworks and tools that can be used to develop blue ocean strategies (Kim and Mauborgne, 2013a).

Both the competitive strategy and the blue ocean strategy stress the importance of avoiding intense competition and acknowledge that innovation increases company profitability. But the blue ocean strategy is more optimistic about the impact of innovation. The competitive strategy assumes that innovation can offer short-term benefits to a business, but competitors quickly imitate the innovation and erode the profits. The blue ocean strategy posits that with proper “value innovation” companies will be able to continuously discover untapped blue oceans, create customer demand for them, and grow without direct confrontation with competitors (Burke et al., 2016).

A study by Burke et al. (2016) looked at the Dutch retail industry over the span of 18 years. Over that period, the number of brands and differentiation strategies in that industry grew, increasing market segmentation, growing market boundaries, and rejuvenating stalling sectors. The researchers found a long-term positive relationship between the number of companies and average profits. According to the competitive strategy, an increased number

of firms in an industry would negatively impact average profits. The study, however, found that while the number of companies grew, the average profits also increased. This indicates that there was a sufficient amount of untapped market opportunities within the industry to avoid the negative impacts of competition (Burke et al., 2016).

Strategy Canvas is one of the tools in the blue ocean toolkit (Kim and Mauborgne, 2013c). The use of the Strategy Canvas for product strategy development is suggested by Pichler (2016). An example of the canvas is presented in Figure 8. The X-axis lists the competing factors in an industry. This can include features, design, price, and other categories. The Y axis specifies the offering level that buyers receive. The tool allows companies to evaluate the current state of the market space, and what the buyers expect from the industry. This is referred to as the industry value curve, or as the “red ocean reality” in blue ocean literature. Next, the new product can be evaluated against the same competing factors to showcase how it will stand out. A product whose curve is too close to the industry value curve is not sufficiently differentiated, and it may be hard to justify to the customers why they should prefer it over the competition (Kim and Mauborgne, 2013c; Pichler, 2016).

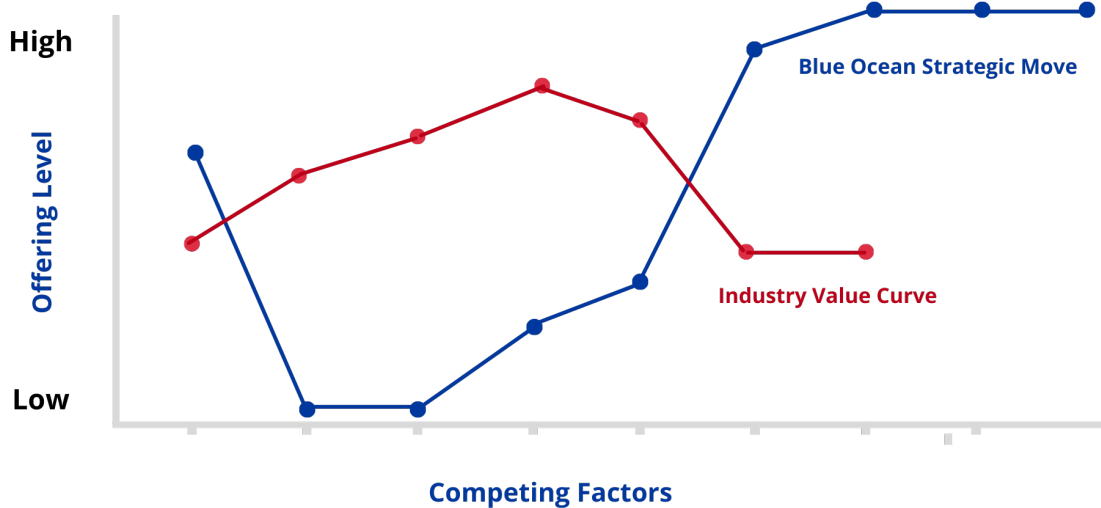


Figure 8. Strategy Canvas (Kim and Mauborgne, 2013c)

An example in Figure 9 compares the 2007 iPhone to the mobile phone industry of the time.

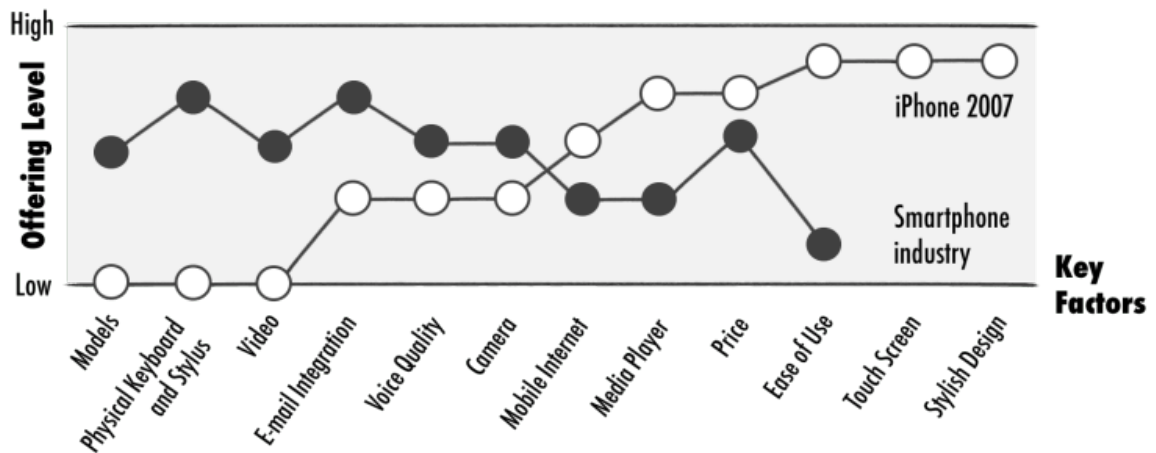


Figure 9. Strategy Canvas for the first iPhone (Pichler, 2016)

To differentiate the product from the industry value curve, competing factors need to be removed, reduced, elevated, or created. Another tool from the Blue Ocean kit helps to make these decisions. This is the Eliminate-Reduce-Raise-Create Grid, presented in Figure 10. The tool encourages making informed trade-offs, instead of blindly trying to implement every factor of the competition, but better (Pichler, 2016). The Grid aims to push companies to achieve both differentiation and low cost by focusing on the factors that make the product stand out and removing unnecessary factors as clutter (Kim and Mauborgne, 2013b).

ELIMINATE	RAISE
Which factors that the industry has long competed on should be eliminated?	Which factors should be raised well above the industry's standard?
REDUCE	CREATE
Which factors should be reduced well below the industry's standard?	Which factors should be created that the industry has never offered?

Figure 10. The ERRC grid (Kim and Mauborgne, 2013b)

The Strategy Canvas can be combined with the ERRC Grid. Looking back at Figure 9, one can see that the value curve for the iPhone is sorted by the level of offering, from low to high. The Canvas shows that the iPhone eliminated certain competing factors when compared to the smartphone industry of the time, offered certain factors to a smaller extent, improved on certain factors, and offered some brand-new factors, uncontested and

unmatched in the industry. The competing factors that are created and do not meet any competition are the desired blue ocean, the value innovation which brings the company to the uncontested market space (Pichler, 2016).

An online chart editor Visual Paradigm offers a free template for the Strategy Canvas (Visual Paradigm, 2022). For examples of using the template, refer to Sections 4.2.3 and 5.2.3 of this thesis.

2.8.4. Domain model

A domain model is “an object model of the domain that incorporates both behaviour and data (Fowler, 2015)”. It defines domain abstractions, including objects, relationships, and procedures. A domain model can be represented using a UML class diagram (Sommerville, 2011). The domain model was selected in this thesis as one of the tools to define the business architecture of the product. Recalling the definition in Section 2.5.1, business architecture overlaps with software architecture and may describe the logical data model, the process model, and the business object model for the software. The domain model allows mapping the main abstractions in the domain, their functionality, and the relationships between them. As such, it is suitable for describing business architecture. While the details of the UML syntax are out of scope for this thesis work, here it is possible to define the types of relationships that can exist between entities in a domain model.

- Inheritance (also known as generalization) – between a more general class (Class 1) and a more specific class (Class 2), directed from Class 2 to Class 1 that it extends (Visual Paradigm, 2019).
- Realization / Implementation – the relationship between the interface and the implementing class, directed from class to the interface (Visual Paradigm, 2019).
- Aggregation – a “part of” relationship, used when multiple instances of one class (Class 2) can be associated with another class (Class 1). Unlike in a “Composition” relationship, instances of Class 2 can also exist without association with Class 1 (Lucid Software, 2017; Visual Paradigm, 2019).
- Composition – a type of aggregation where one class cannot exist without another. If Class 2 has a composition relationship with Class 1, instances of Class 2 are

destroyed when an instance of Class 1 that Class 2 is composed with is destroyed (i.e., parts cannot exist outside of the whole) (Lucid Software, 2017; Visual Paradigm 2019).

- Association – a relationship between classes (Visual Paradigm, 2019). Associations have a verb describing the association on top of the link in the diagram. (Visual Paradigm, 2019).
- Dependency – an instance of one class uses an instance of another class (Visual Paradigm, 2019).

Each relationship is represented by a different type of arrow in a UML class diagram. Figure 11 presents the types of arrows for each relationship.

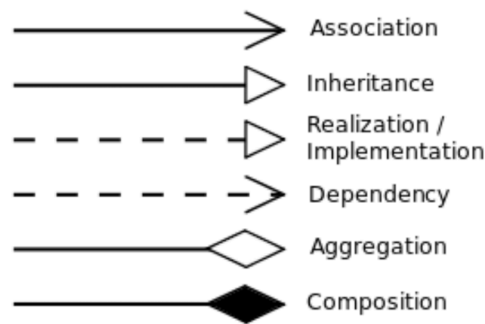


Figure 11. Relationship types in a UML class diagram

Multiplicities can be specified in the diagram above the arrows connecting two entities, specifying the number of instances on both sides of the relationship.

2.8.5. Business Process Model and Notation (BPMN)

Business Process Model and Notation is a standard notation for describing business processes. It defines a set of events, activities, and gateways, that can be arranged in swim lanes to describe business processes within a company, and the interactions between business units within an organization and with external parties (OMG, 2011). The details of the BPMN syntax are out of scope for this thesis, but the official documentation provides a quick guide for getting started (see OMG, 2022). Figure 12 presents an example of a BPMN model.

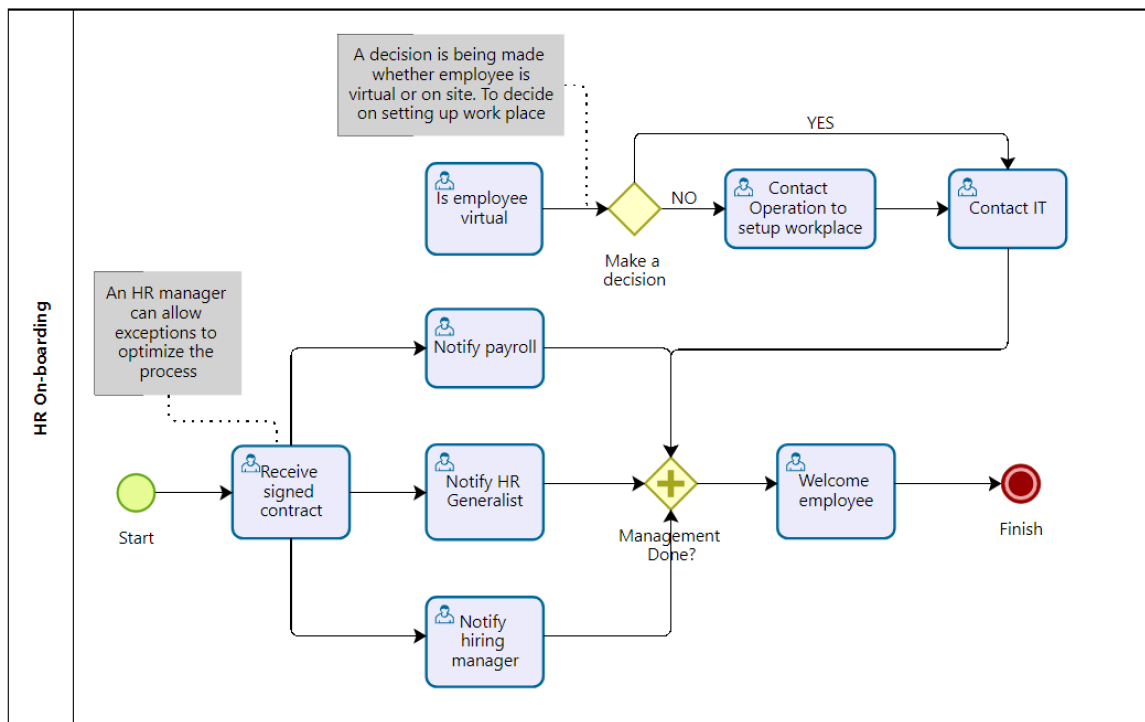


Figure 12. An example BPMN diagram for an imagined business process

Multiple tools exist for creating models in BPMN notation. One such tool is the Bizagi Modeler. Using the tool, BPMN concepts (events, tasks, gateways, pools, lanes, and data objects) can be dragged from a menu onto an interactive canvas, where they can be arranged as needed. The tool is available for free for personal use and offers several paid tiers. The paid professional version of the modeller includes a simulation feature, where the business processes can be validated in terms of their duration, resource efficiency, and performance under constraints such as shift work and holidays (Bizagi, 2022).

BPMN was selected in this thesis for defining parts of the business architecture related to business processes.

2.8.6. Accion pricing framework

Accion is a non-profit financial consulting company that offers a SaaS pricing framework (Accion, 2022). The framework proposes a 4-step process to set the price for a SaaS offering. Accion framework was included in a study by Saltan and Smolander (2019) that compared and analysed academic and practitioner literature about SaaS pricing. The study did not find a universal SaaS pricing cookbook but identified 13 SaaS pricing frameworks. The paper

concludes by calling for further practical use and evaluation of the frameworks. The effectiveness, ease of use, and adaptability of the frameworks need to be analysed. The Accion framework was selected in this thesis due to its action-oriented approach and clear process. The latest edition of the ISPMA-compatible study guide and handbook (Kittlaus, 2022) also suggests the Accion framework as a practical decision support tool for price setting.

Accion provide their framework in the form of a slide deck (Accion, 2015). The slide deck begins by introducing value-based, competitor-based, and market-based pricing, as illustrated in Figure 13. The framework suggests predominantly using value-based pricing, as it maximizes the potential revenue for the SaaS vendor. The three types of pricing are illustrated using a Picasso painting as an example.

Pricing should be based on value to customer

Cost-Based Pricing	Competitor-Based Pricing	Value-Based Pricing
Use Rarely	Use Sparingly	Use Predominantly
<ul style="list-style-type: none"> ✓ Revenues will cover costs ✓ Requires little research to set 	<ul style="list-style-type: none"> ✓ Prices are in-line with customers' past expectations and vocabulary ✓ Direct comparability exists between players (advantageous in an RFP process) 	<ul style="list-style-type: none"> ✓ Maximizes revenue by charging each customer maximum will pay
<ul style="list-style-type: none"> ✗ One price for all customer segments ✗ Undercharges customers willing to pay more ✗ May not stay profitable if costs change 	<ul style="list-style-type: none"> ✗ Competitors may not exist for an innovative product or service ✗ Can create a "race to the bottom" between competitors, eroding margins 	<ul style="list-style-type: none"> ✗ Requires extensive research and experimentation to refine ✗ Charging too high a price may create margins that tempt entry of more competitors ("price umbrella")
Virtually never the correct strategy because it undercharge customers willing to pay more, and because your margins may be unsustainable if your cost structure changes	Sometimes appropriate for very mature markets with established price points, but usually not the best strategy because it undercharges customers willing to pay more	Nearly always the best strategy because it secures the greatest possible revenue for a product

Figure 13. Pricing approaches and their trade-offs (Accion, 2015)

The framework then goes through the four steps.

1. Defining the upper bound.

Here, the maximum amount that can be charged to customers is determined. This depends on several factors related to the competitive advantage and the customer's willingness to pay:

- a. The uniqueness of the product. Vendors of unique products can act as “price-makers”. Such products stand out with their unique functionality, excellent customer service, brand perception, or geographic focus, for example. If a product is similar to other products, the price is the only significant differentiator, and the customer will always prefer the cheapest available option (Accion, 2015).
- b. The contribution of the product to the customer's bottom line. The product may increase revenue for the customers and/or decrease costs (Accion, 2015).
- c. Switching costs for the customers. The framework argues that high switching costs are a double-edged sword for SaaS companies, as they can complicate the adoption of the software, but later prevent the customers from leaving. The framework suggests minimizing the adoption costs for the customer and enhancing retention. Adoption can be made easier by freemium models, migration support, training materials, and product ease of use. Retention can be achieved by excellent customer experience, development of trusting relationships, and network benefits that the customer gains when more customers start using the SaaS product (Accion, 2015).
- d. Key customer segments and their price sensitivity. The framework suggests identifying key customer segments, which can be differentiated by demographics, needs, or behaviours. Customer and market research needs to be conducted to make informed segmentation decisions. Thought-out segmentation allows to charge higher prices to less price-sensitive segments, prioritize less price-sensitive segments, and increase competitive advantage by offering functionality especially valuable for certain segments. Segmentation also helps marketing and advertising use the right channels for each segment (Accion, 2015). The framework lists factors that influence customer price sensitivity.
- e. Understanding of the value by the customer. There may be a gap between the value the product brings, and the value perceived by the customers. Customers may be sceptical about the product, or reluctant to migrate from their established workflows. The true value of the product only reveals itself once the customers use it for a certain period.

Accion view to the time it takes to understand the true value of the product as an expense that needs to be incorporated into the initial price. Innovative or complex products can be offered with a discounted or free trial period (Accion, 2015).

Based on these factors, the framework offers a table to calculate an estimate for the upper bound. The table is presented in Figure 14. Each row addresses one of the factors. Depending on the context, each row might have a positive, negative, or neutral implication on the maximum product value. Numeric values can be assigned to positive and negative implications. For example: “The product increases loan officer productivity and allows the customer to eliminate one full-time position. This saves the customer 20000\$ a year on average. The pricing implication is therefore 20000\$ (Accion, 2015)”.

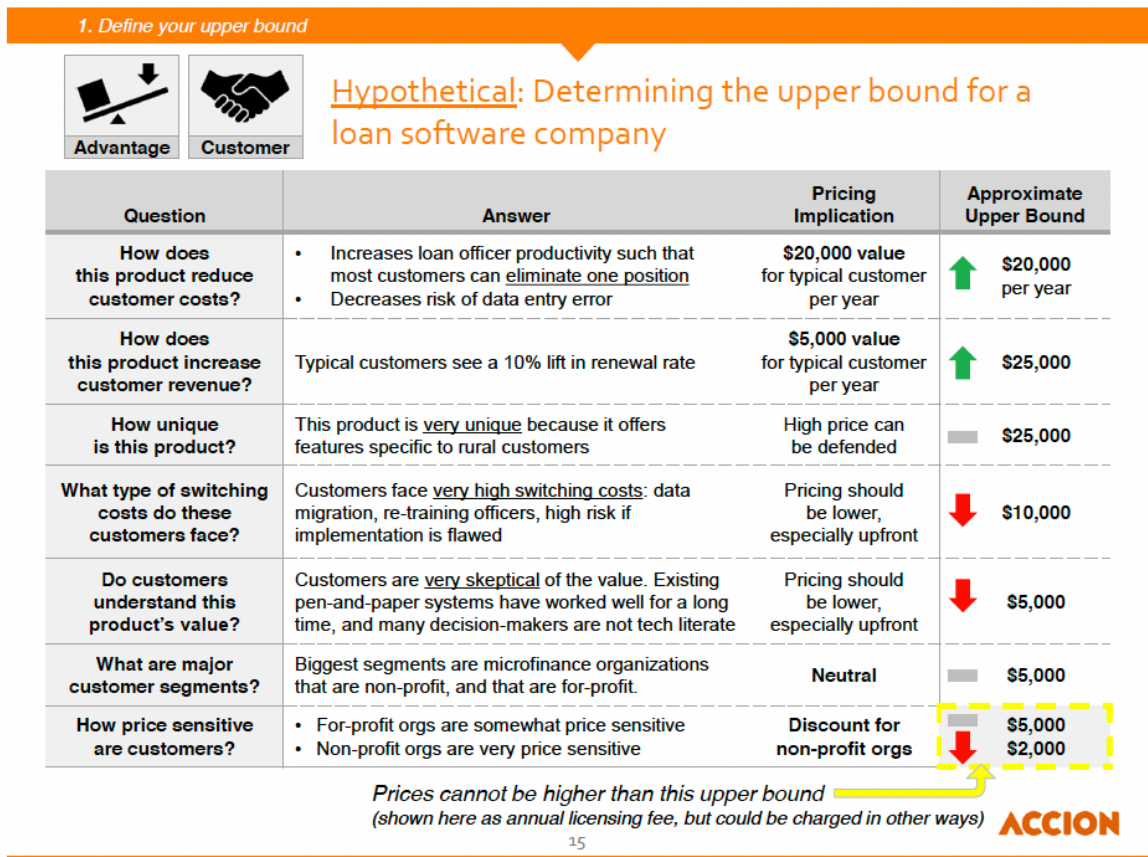


Figure 14. Upper bound calculation (Accion, 2015)

Table 4 is a blank template for the upper bound calculation. The template was made for this thesis based on the Accion table in Figure 14.

Table 4. Upper bound calculation template (adapted from Accion, 2015)

Question	Answer	Pricing implication	Approximate upper bound
How does this product reduce customer costs?			
How does this product increase customer revenue?			
How unique is this product?			
What type of switching costs do these customers face?			
Do customers understand this product's value?			
What are major customer segments?			
How price sensitive are customers?			

The template in Table 4 can be used to make upper bound estimations for a SaaS product.

2. Defining the lower bound.

During this step, the minimum revenue is calculated that is necessary for the product to offset fixed and variable costs and become profitable. Common sources of fixed and variable costs for SaaS products are listed. Fixed costs include executive compensation, product development, legal approval of products, and certifications and registrations. Variable costs include data storage and server space, advertising, sales commissions, customer service, licenses of third-party components, and insurance (Accion, 2015).

3. Identifying any reasons to charge less than the maximum value.

At this point, any reasons to charge less than the upper bound are identified. The three general reasons for charging less are competition, discounts, and value metrics. Describing the competition factor, Accion mentions a few cases when a SaaS company can consider undercutting the price of their competitors. The reasons for undercutting a competitor may be:

- a. SaaS company has a cost structure that competitors cannot match
- b. SaaS company aims to take share from an established business
- c. SaaS company operates at a high volume that enables unit economics that is cost-prohibitive for new entrants (Accion, 2015)

Multiple reasons for discounting are listed, including discounting for high-volume customers, encouraging customers that refer other customers, charging less from customers that commit for longer periods of usage, and bundling less desirable products with more desirable ones (Accion, 2015).

Finally, the concept of value metrics is described as “the increment of your product for which you charge for increased access”. Examples given are accounts, bandwidth, storage, access to features, and customer support. Best practices for the selection of value metrics are provided. The metrics shall be closely tied to the customer segmentation to offer maximum value for each segment. The metrics should also be communicated to the customer in an understandable way (Accion, 2015).

4. Structuring the pricing model as a compromise between the upper and lower bounds.

At this stage, three common SaaS pricing structures are described, including freemium, consumption-based, and tiered. The fourth type, perpetual license, is also mentioned as rarely used for modern SaaS. Justifications for using each model are provided. It is mentioned that SaaS companies can combine these price structures. It is also possible to charge separately for installation and onboarding. Such additional costs must be communicated upfront, and preferably incorporated into the recurring price instead (Accion, 2015).




The framework concludes by showing screenshots of real SaaS pricing pages. The screenshots are annotated by Accion. Value metrics, price structures, and types of discounts are pointed out.

2.8.7. Balanced Product Scorecard

The concept of tracking business performance using a balanced scorecard was already introduced in Section 2.5.8. A Balanced Product Scorecard is a variation of BSC described by Pichler in his book “Strategize” (2016). The framework is presented in Figure 15. This variation of the scorecard lists the business goals at the top, as a way to ensure only the KPIs relevant to the business goals are considered.

THE BALANCED PRODUCT SCORECARD

romanpichler

 BUSINESS GOALS <p>The desired business benefits the product should create, prioritised and measurable.</p>	
 FINANCIAL <p>The relevant financial indicators, for instance, revenue, cost, profit.</p>	 CUSTOMER <p>The relevant customer indicators, such as market share, adoption rate, referrals.</p>
 PRODUCT AND PROCESS <p>Product and process indicators including product health and schedule variances.</p>	 PEOPLE <p>People indicators, like team motivation and stakeholder engagement.</p>

www.romanpichler.com
Template version 10/16

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License




Figure 15. Balanced Product Scorecard (Pichler, 2016)

Pichler also lists sample KPIs for each of the four perspectives. The full list of these sample KPIs is presented in Appendix 1.

3. Design research problem and artefact requirements

The thesis follows the design science research process as described by Johannesson and Perjons (2021). The visual representation of the process is displayed in Figure 16. The five stages of the process are described by the authors as follows:

1. **Explicate Problem.** In this phase, the problem that the research is trying to solve is analysed and formulated precisely. A question answered during this phase is “what is the problem experienced by some stakeholders of a practice and why is it important?”
2. **Define Requirements.** In this phase, the requirements for the developed artefact are stated. The question to answer is “What artefact can be a solution for the explicated problem and which requirements for this artefact are important to the stakeholders?”
3. **Design and Develop Artefact.** In this phase, an artefact is created.
4. **Demonstrate Artefact.** In this phase, the artefact is applied to a specific case to demonstrate its feasibility. The question to answer in this phase is “How can the developed artefact be used to address the explicated problem in one case?”
5. **Evaluate Artefact.** This phase reflects on how well the artefact was able to solve the explicated problem and fulfil its requirements.

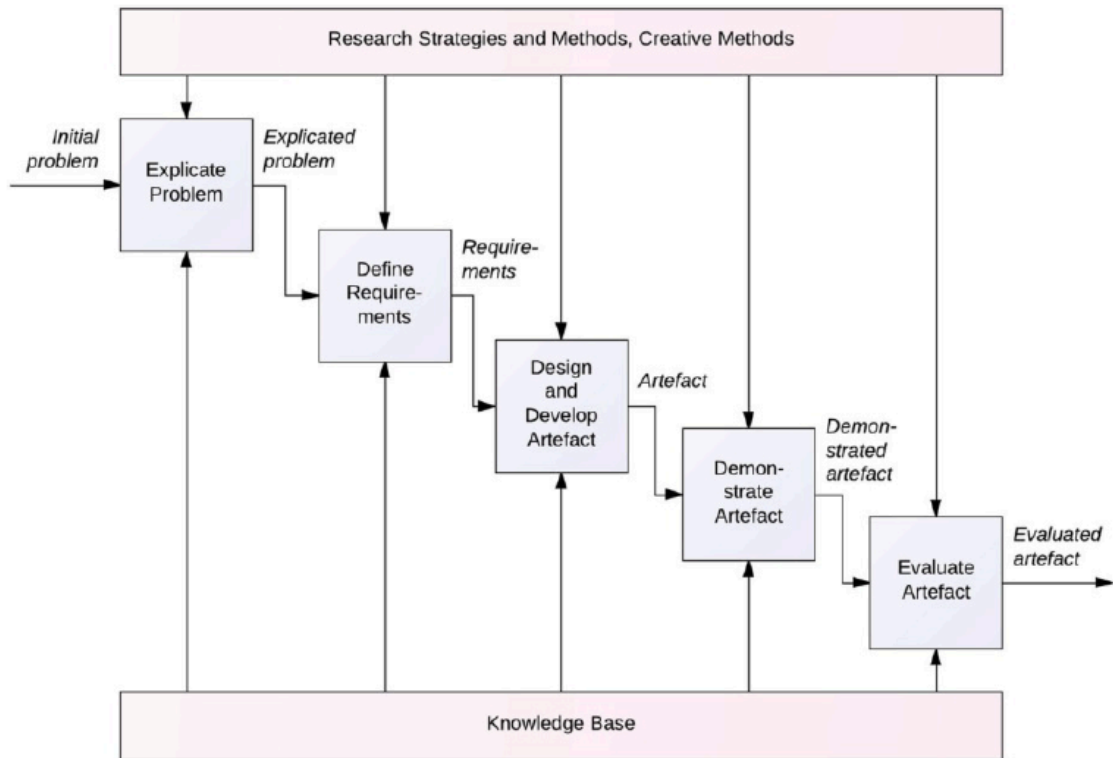


Figure 16. The method framework for design science research with research strategies and knowledge (Johannesson and Perjons, 2021)

The thesis follows the guidelines for design science introduced by Hevner et al. (2004). The guidelines were developed to clarify the criteria of high-quality design research (Johannesson and Perjons, 2021). The framework by Hevner et al. (2004) is presented in Figure 17. The framework places the generated theories and artefacts in the context of the surrounding business environment and the scientific knowledge base. The environment is the source of business needs that the novel artefact will address. The artefact is created in response to the business needs and is then applied to address the business needs within the environment. The strong interconnection between the artefact and the environment ensures the relevance of the artefact. The right side of the framework grounds the creation of the artefact in the existing knowledge base. The decisions made during artefact development are based on established theories and models. The resulting artefact is a source of new knowledge, that itself contributes to the knowledge base. The strong interconnection between the artefact creation and evaluation and the knowledge base indicates the appropriate level of scientific rigour was exercised when developing the artefact (Hevner et al., 2004).

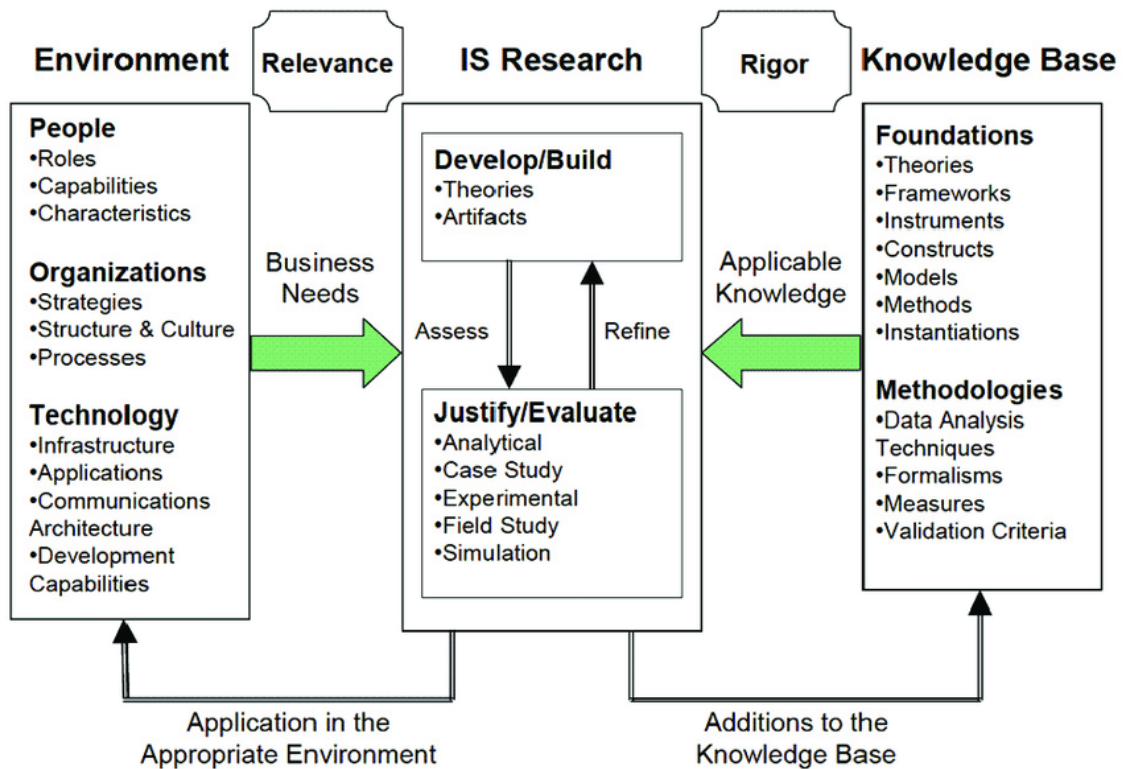


Figure 17. Information Systems Research Framework (Hevner et al., 2004)

In the context of this thesis, the environment is the commissioner company. The company is a software vendor that faces a real business need – the need for a clear product strategy for a B2B SaaS product, which is a productized version of an existing customer specific software system. The product is based on the technologies developed by the commissioner company over the past several years, as well as the development capabilities to adapt those technologies into a new SaaS product. The knowledge base is the established scientific understanding of software product strategies, productization of customer-specific software, and the characteristics of B2B and SaaS products. The ISPMA SPM framework and the tools described in Section 2.8 are used for the development of the strategy for the new product.

The artefact produced in this thesis is designed to solve the problem of the commissioner company. Design science calls for solving practical problems of general interest using novel artifacts (Johannesson and Perjons, 2021). Solving a practical problem for the commissioner company alone would fail to meet the “general interest” part of this definition. The artefact therefore was designed in a way that could make it useful to other software development organizations wishing to productize their customer-specific offerings and improve their product management activities. In this way the artifact aspires to contribute to the knowledge base – the SPMBOK. During the research process, knowledge about the artefact is

accumulated, including the influence of the B2B SaaS context and the productization context on product strategy decisions.

3.1. Explicating the problem

Kittlaus and Fricker (2017) write that product strategy development is a continuous process, which spans the lifecycle of a product and consists of multiple disjointed activities. Because of this, the authors argue, defining a coherent process for product management, and specifically product strategy, “does not make a lot of sense” (Kittlaus and Fricker, 2017, p.116). This, however, leaves a product manager charged with developing a strategy for a new product in a perplexing position. They need to develop a comprehensive product strategy, considering all 8 elements of the strategy column of the SPM framework, and working in close collaboration with Marketing, Sales, and executive management (Kittlaus and Fricker, 2017). Certainly, some structure is welcome for the initial strategy development. It is, for example, futile to start with KPI selection for performance and risk management before formulating a problem and position statement.

Moreover, while the final documented product strategy is important, it is primarily a tool that is used to discuss key decisions, consider crucial details, elicit feedback from stakeholders and formulate a shared and accepted plan (Bavaro and McDowell, 2021). According to Bavaro and McDowell (2021, p.104), “the final deliverable isn’t as valuable as the process you go through to write the documentation”. The following problem can be formulated:

“Product management is seen as a continuous activity with many separate tasks, and no ordered process exists to guide a product manager in initial strategy development”.

The term “initial strategy development” is used to emphasize that the thesis does not claim that it is possible to determine a complete and final strategy in one execution of a sequential process. The thesis only claims that an ordered process can help a product manager learn what areas of product strategy they need to develop, create an initial understanding of the strategy, and the activities that need to be routinely revisited over the product lifecycle.

3.2. Defining artefact requirements

The following requirements have been identified for the process model.

1. The model shall provide direction to a product manager in charge of establishing a strategy.
2. The model shall lead to the creation of a product strategy.
3. The resulting strategy must describe all 8 elements of the “Product strategy” column of the ISPMA SPM framework.
4. The resulting strategy shall be applicable to a B2B SaaS software product.
5. The model shall suggest methods and tools that are effective for B2B SaaS strategy development.
6. The model shall utilize company resources efficiently by ensuring only the necessary stakeholders are required to attend certain phases.
7. The resulting strategy is documented in a single product strategy document, which can be used to communicate the strategy across the organization.

Figure 18 displays the design research process overall. The completed activities of the process are marked in green.

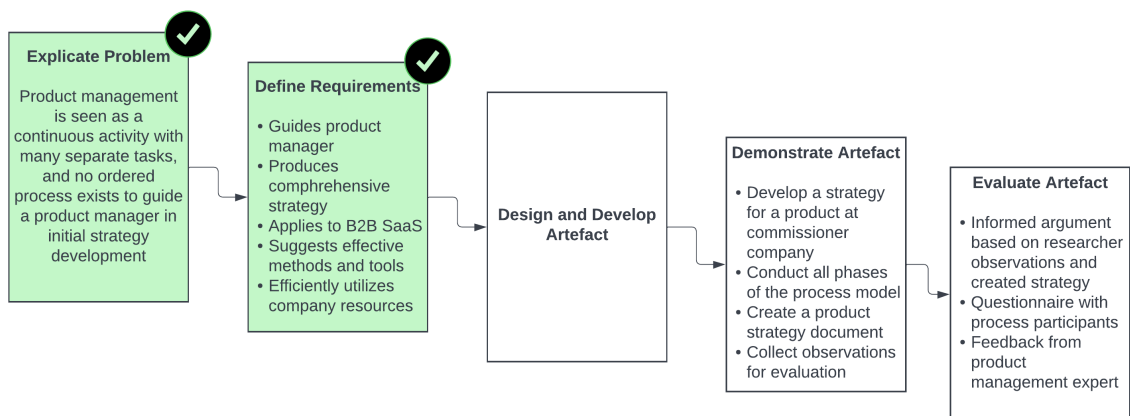


Figure 18. Design research process after explicating the problem and defining requirements

Having explicated the problem and defined requirements, it is now time to design and develop the artefact.

4. Model for initial strategy development for a B2B SaaS product

This chapter presents the proposed process model for initial product strategy development, supporting productization of customer-specific software. This chapter corresponds to the “Design and develop artefact” step of the design research process by Johannesson and Perjons (2021). The process model contains 6 phases. In Section 4.1 the overall model is discussed. A visual representation is provided. In Section 4.2 each phase is described in more detail. The design decisions made when creating the process model are based on the theoretical background laid out in Chapter 2.

4.1. Overview of the process model

Figure 19 presents the proposed process model for initial product strategy development supporting productization of customer-specific software. The boxes at the centre of the model are the 8 elements of product strategy according to the ISPMA SPM framework. The bubbles around the boxes indicate the phases of the process model. The dashed arrows from each bubble to the next indicate the order in which the phases should be executed.

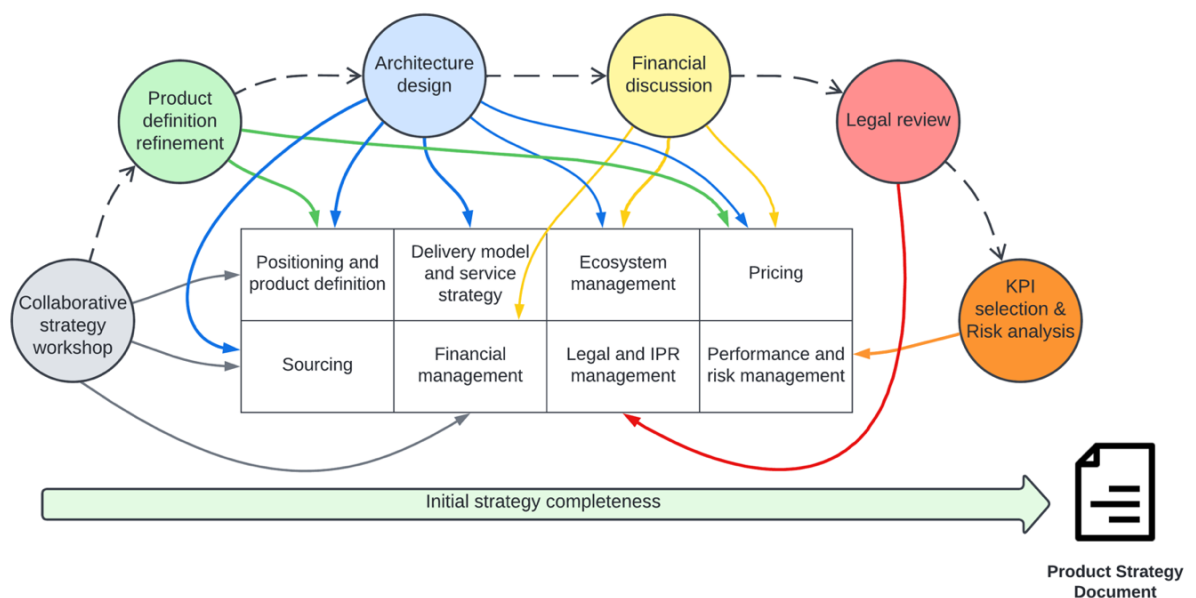


Figure 19. Process model for initial product strategy development for B2B SaaS

Each bubble is linked to one or more elements of product strategy. For example, the “Product definition refinement” phase (the green bubble) is linked to “Positioning and product

definition” and “Pricing” elements, indicating that these elements are refined during this phase of the process. Each bubble and its arrows are colour-coded to simplify the visual comprehension of the model. The process model is described in more detail in Table 5. For each phase, the key questions that need to be answered during the phase are specified, alongside the recommended tools and the strategy elements impacted during the phase.

Table 5. Phases of the process model with key questions, tools, and impacts

Collaborative strategy workshop¹			
Questions		Tools	Impacts
What is the motivation to create the product? What positive change will it bring?	Who are the competitors? What are their strengths and weaknesses?	Product Vision Board	Positioning and product definition
Who are the target users? What are the market segments?	How will the product be monetized?	Problem and position statement template	Sourcing
What problem will the product solve?	What are the main cost factors to develop, market and sell the product?		Financial management
What is the product? What makes it stand out?	How will the product be marketed and sold? What channels are needed to reach customers?		
How will the product benefit the company? What are the business goals?			
Product definition refinement			
Questions		Tools	Impacts
What features will the product include? What quality attributes will the product possess?		Blue Ocean Strategy Canvas	Positioning and product definition
How will the product compare to the competition in terms of functionality, user experience, quality, and price?		Blue Ocean Eliminate-Reduce-Raise-Create (ERRC) Grid	Pricing
Can the product offer some feature or attribute that currently doesn't exist in the market?			

¹ Questions in this phase are based on the Product Vision Board template by Pichler (2016)

Architecture design			
Questions		Tools	Impacts
1. What is the defining technology for the software product? What technology enables our competitive edge and market differentiation over time?	3c. How customizable do we want the software to be? (With SaaS this is rare) Are we willing to modify the software code for specific customers?	UML Domain Model BPMN business process models	Positioning and product definition: offering architecture, business architecture
2. What is the offering architecture? Meaning what are the separately priced components of the product?	4. What is our desired place in a software ecosystem? What organizations could we partner with? What external systems could we integrate? Could we provide a way for third parties to integrate into our software?		Delivery model and service strategy, Tailorability
3. What is the tailorability architecture? Meaning:	5. What is the business architecture?		Ecosystem management
3a. How configurable do we want the software to be? What parameters are configurable?	5a. What is the domain model of the new software?		Pricing
3b. How composable do we want the software to be? What components can be added or removed?	5b. What business processes need to be created to support the new product?		Sourcing
Financial discussion			
Questions		Tools	Impacts
Is there any reason to choose cost-based or competitor-based pricing over value-based pricing?	Are there any reasons to charge less than the maximum value?	Accion pricing framework	Pricing Financial management
What is the upper pricing bound? What is the maximum value the product has for customers?	What will the pricing structure be for the product? What combination of freemium, consumption-based, and tiered pricing can the product have? Is a perpetual license an option?		
What is the lower pricing bound? What are the fixed and variable costs for the product?			

Legal review			
Questions		Tools	Impacts
Contracts. Who is responsible for formulating service contracts in the organization? What are the terms of the service level agreement (SLA)? Does the organization have templates for such SLAs? Can any existing SLAs be reused?	Open source. What open-source components may be used when developing and running the software? What are the distribution licenses for those components? Are there any restrictions or caveats?	Checklist of legal aspects to review	Legal and IPR management
IPR protection. How will the company protect the intellectual property rights related to the product? Does the company have patents or trademarks that apply to the product? Should the company obtain new trademarks or patents for the product?	Who is responsible for formulating the privacy policy within the organization? What are the terms of the policy? Does the organization have templates for such policies? Can any existing policies be reused?		
KPI selection and Risk analysis			
Questions		Tools	Impacts
What are the business goals of the company? How can they be measured? What targets should we set for those goals?		Balanced Product Scorecard	Performance and risk management
What financial, customer, product and process, and people KPIs should we track to achieve the business goals? How should those KPIs be measured?		“Digital Red Dot Game”	
What elements of the overall product strategy are we least certain about?			

Figure 19 depicts the process in a linear, sequential way. But this does not imply that the strategy produced at the end of the process is complete and is never to be changed. The term “**initial** product strategy” is used to emphasize this. The phases of the process model can and must be revisited routinely when more information becomes available or any relevant product goals or decisions are changed. Possible modifications to the description of the process model that highlight an iterative nature of the strategy design process are considered in the “Discussion” chapter. The rest of this chapter describes each of the phases of the process model.

4.2. Phases of the process model

In this section, each phase of the process model is discussed. Each phase is presented using the following template:

Name: The name of the phase.

Impacted elements of product strategy: Each phase of the process model clarifies the contents of some of the 8 elements of product strategy classified in the ISPMA SPM framework. This section lists which elements of the product strategy are discussed in this phase.

Tools: The tools used by the product manager to guide the product strategy discussion in this phase of the process model.

Participants: Members of the organization that need to participate in this phase. The product manager participates in every phase and is therefore not mentioned explicitly.

Approach: This describes succinctly the approach taken by a product manager to conduct this phase. The approach section provides a summary of the phase. The approach is defined in detail in the following sections: preparation, process, produced deliverables, and post factum analysis.

Preparation: This describes how a product manager should prepare for this phase.

Process: This describes what happens during the phase, how the phase is conducted.

Produced deliverables: This describes which deliverables (e.g., documents, sketches, diagrams) need to be produced at the end of the process.

Post factum analysis: At the end of the phase, the product manager needs to analyse and summarise the discussions that occurred during the phase and the deliverables produced. Based on these, the product manager fills certain sections of the product strategy document.

Key design decisions: The main design decisions made when creating the phase. This can be the selection of the tools for the phase, the structure of the phase, or any other aspect.

4.2.1. Collaborative strategy workshop

Name: Collaborative strategy workshop

Impacted elements of product strategy: Positioning and product definition, sourcing, financial management

Tools: The Product Vision Board, Problem and position statement template

Participants: Representatives from executive management, Development, Sales and Marketing.

Approach: Key decision makers are gathered to discuss an initial vision for the product. The aim is to establish a common understanding, identify possible disagreements and reach a consensus.

Preparation: In preparation for the meeting, the product manager sends out the Product Vision Board to all stakeholders and asks them to think about their responses in advance of the meeting. This gives participants time to formulate their initial ideas.

Process: During the workshop, the product manager introduces the Product Vision Board again, first overall, and then each section of the board. After each section is introduced, participants are requested to share their ideas related to the section. It is possible to go around the table to ensure every participant contributes. The product manager participates in the discussion, listens to the stakeholders, and takes notes.

Produced deliverables: The notes taken by the product manager. It is possible to write the notes directly onto the Product Vision Board or write the notes in a text document that follows the same structure as the Product Vision Board.

Post factum analysis: After the meeting, the product manager formulates a problem and position statement according to the template described by Kittlaus and Fricker (2017).

Key design decisions: Using the Product Vision Board instead of the more well-known Business Model Canvas or Lean Canvas. The Product Vision board, while less detailed than those alternatives, was deemed sufficient to elicit initial ideas.

4.2.2. Product definition refinement

Name: Product definition refinement

Impacted elements of product strategy: Positioning and product definition, (possibly) Pricing

Tools: Strategy Canvas, Eliminate-Reduce-Raise-Create grid (both tools are part of the Blue Ocean family of tools). A free online Strategy Canvas editor is offered by Visual Paradigm (Visual Paradigm, 2022).

Participants: Project manager, Sales representative

Approach: At this phase of the process, the product definition is enhanced with more details. The goal is to determine which features and quality attributes the new product will offer. The new product is compared to existing market offerings to determine what features or attributes can be created, improved compared to competitor offerings, reduced compared to competitor offerings, or eliminated.

Preparation:

1. The product manager analyses identified competitor products. The product manager looks at the existing market offering, as well as the software that is being productized into B2B SaaS. The product manager picks features and quality attributes (referred to as “competing factors” from now on) that are available in the market and/or are already offered in the existing software, as well as new competing factors that can be provided by the new product.
2. The product manager prepares the online Strategy Canvas Tool. Competing factors are added to the first column of the table (Figure 20). The second and third columns list the level at which the factors are offered by the industry, and by the new product. The initial score for each competing factor can be set to zero. This input results in a Strategy Canvas that looks like in Figure 21.

	A	B	C
1		Industry	New product
2	Factor 1	0	0
3	Factor 2	0	0
4	Factor 3	0	0
5	Factor 4	0	0
6	Factor 5	0	0
7	Factor 6	0	0
8	Factor 7	0	0
9	Factor 8	0	0
10	Factor 9	0	0
11	Factor 10	0	0
12			

Figure 20. Data entered into the online Strategy Canvas tool at the beginning of the phase

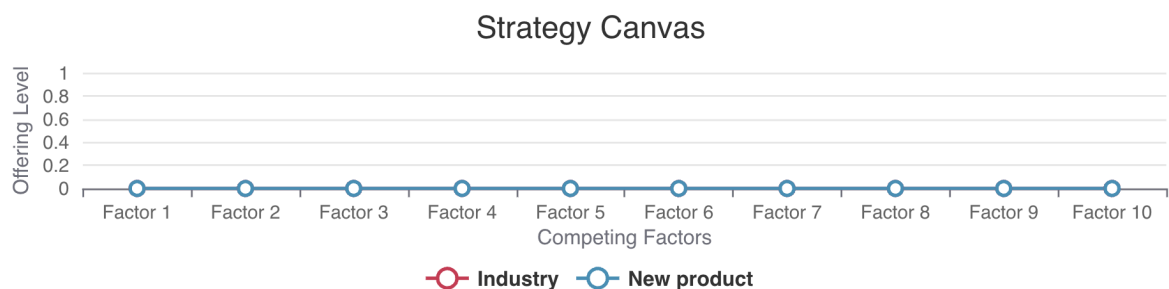


Figure 21. Strategy Canvas before the meeting

Process:

1. The product manager introduces the Strategy Canvas tool.
2. The group estimates how features will be scaled. A 1 – 4 scale is recommended since all the estimates are approximate and having a larger scale may not provide additional value. But this can be adjusted based on the needs of the company. For example, it is possible to increase the possible maximum score to highlight that the new product will be miles ahead of the competition.
3. During the meeting, the group goes through each category and estimates the industry offering and the desired product offering.
4. The product manager asks the participants whether any relevant categories are missing from the list.
5. The product manager asks the participants whether some of the categories can be eliminated from the new product or offered at a reduced level.

- The product manager listens for any possible new categories mentioned by the participants and adds them to the canvas if deemed relevant.

Produced deliverables: The filled Strategy Canvas with scored competing factors, looking similar to the Canvas in Figure 22.

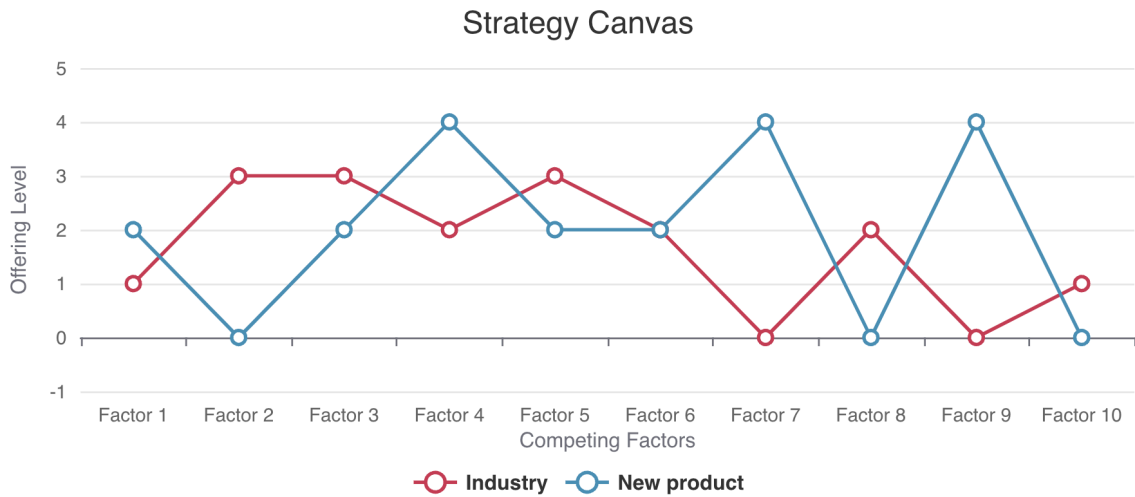


Figure 22. Strategy Canvas at the end of the meeting

Post factum analysis: The product manager sorts the factors by their score from low to high. This essentially combines the strategy canvas with the ERRC grid. The top right corner indicates the potential Blue Ocean for the product – competing factors that are offered by the product at an excellent level, and not offered by the industry at all. An example is presented in Figure 23.

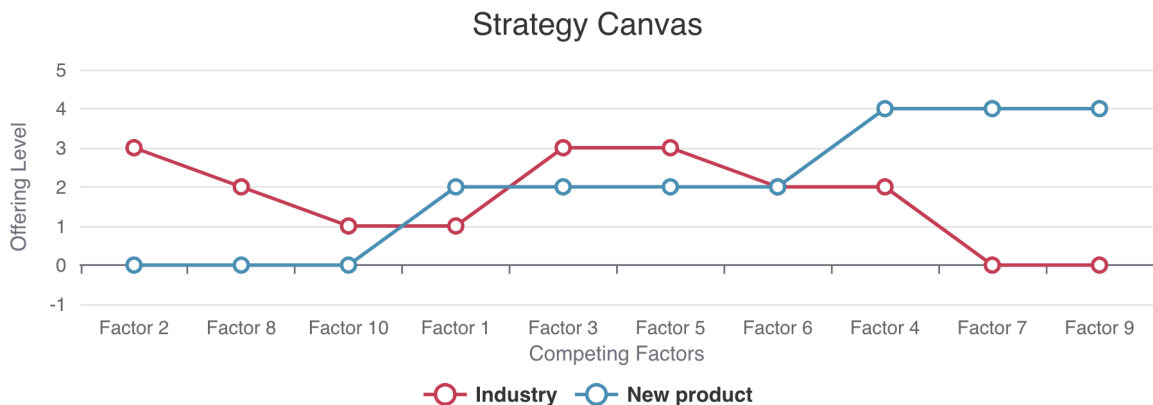


Figure 23. Strategy Canvas combined with ERRC

If there are too many competing factors, it is possible to group these factors into some related categories. For an example of such grouping, refer to the demonstration of the process model in Chapter 5 (Table 7).

Key design decisions:

1. Using the Strategy Canvas from the Blue Ocean family of tools.
2. Compiling the list of competing factors for the Strategy Canvas based on both the possible competing products and the existing software system that is being productized.

This concludes the product definition refinement phase. The architecture design phase is discussed next.

4.2.3. Architecture design

Name: Architecture design

Impacted elements of product strategy: Positioning and product definition, Tailorability, Ecosystem management

Tools: domain model, BPMN business process models

Participants: CTO (if available), software architect, senior software engineer

Approach: In this phase, the offering architecture, business architecture, and tailorability are discussed. The place of the product in software ecosystems and the possibility of creating ecosystems are touched upon. It is recommended to conduct two meetings during this phase. In the first meeting, overall architecture considerations, the offering architecture, and the place in a software ecosystem are to be discussed. Based on this preliminary discussion, the product manager can prepare a domain model for the new product and relevant business process models. Those models can be presented in the second meeting to elicit further discussion and refine the strategy.

Preparation: The product manager shares the questions with the participants in advance of the meeting. This gives participants time to formulate their initial ideas.

Process: During the first meeting of this phase, the following questions need to be discussed:

1. What is the defining technology for the software product? What technology enables our competitive edge and market differentiation over time?
2. What is the offering architecture? Meaning what are the separately priced components of the product?
3. What is the tailorability architecture? Meaning:
 - a. How configurable do we want the software to be? What parameters are configurable?
 - b. How composable do we want the software to be? What components can be added or removed?
 - c. How customizable do we want the software to be? (With SaaS this is rare) Are we willing to modify the software code for specific customers?
4. What is our desired place in a software ecosystem? What organizations could we partner with? What external systems could we integrate? Could we provide a way for third parties to integrate into our software?

As with the other phases, the goal is to have a discussion with the stakeholders and create a shared understanding. In Chapter 2 it was established that the ISPMA literature does not describe the required contents of business architecture documentation discussion precisely and leaves it up to the organization to determine it based on the specific domain. The question regarding the business architecture requires further clarification. In the context of a B2B SaaS product created during the productization of existing software, the following supplementary questions can be formulated:

5. What is the business architecture?
 - a. What is the domain model of the new software?
 - b. What business processes need to be created to support the new product?

Produced deliverables:

1. Domain model of the new software
2. One or several business process models in BPMN notation

Post factum analysis: The product manager describes the business architecture, offering architecture, tailorability, and software ecosystem elements in the product strategy document.

Key design decisions:

1. Splitting this phase into two meetings. In the first meeting, there is only open discussion around a set of questions. This gives the manager time to process the ideas from the first meeting, visualize them, and refine them in the second meeting.
2. Using a domain model represented by a UML class diagram to create an understanding of the product architecture. To the knowledge of the author, UML diagrams are rarely mentioned or explicitly recommended in the context of product management.
3. Designing only the new business processes that are required to support the new product. The context for this work is productization. The software organization already has established business processes, which it used to develop the software system that is being productized.
4. Before the second meeting, the product manager uses modelling software to create a domain model for the product and BPMN process models for describing new processes that are required to support the new product.

The next phase discusses the pricing and financial management of the product.

4.2.4. Financial discussion

Name: Financial discussion

Impacted elements of product strategy: Financial management, Pricing

Tools: The Accion pricing framework, the slide deck with the framework provided by Accion (Accion, 2015)

Participants: Sales representative, project manager, approval by representatives from executive management

Approach: The Accion pricing framework is followed using the official slides by Accion. Four questions need to be answered as part of the discussion:

1. What is the upper bound?
2. What is the lower bound?
3. What are the reasons to charge less than the maximum value?
4. How to structure the pricing model as a compromise between the upper and lower bounds?

The product manager guides the discussion and takes notes. Based on the discussion, the product manager prepares a draft pricing page for the new SaaS product. This page is presented to meeting participants and representatives of executive management for feedback and approval. Based on the feedback, a second draft of the pricing page is made and accepted as the initial pricing structure for the product.

Preparation:

1. The product manager familiarizes themselves with the Accion framework.
2. The product manager identifies possible value metrics that can be charged for. This can be, for example, the number of users, or certain features.
3. If possible, the product manager prepares the fixed cost estimate for product development. Accounting of the organization can be consulted for this information. In a company that is productizing existing software into B2B SaaS, development costs for previous customer-specific projects can be used as a first estimate.

Process: The product manager recaps the product strategy established so far and introduces the Accion pricing framework. The product manager goes through the official Accion slide deck that describes the framework. The slide deck is structured in four parts, for the four key questions that determine the pricing. The product manager elicits discussion on all four questions and takes notes. After the session, the product manager analyses the notes and proposes a pricing model based on the discussion. The pricing model can then be adjusted based on feedback from the stakeholders. The product manager may have to seek out additional information, such as cost estimates, before formulating a pricing model proposal.

Produced deliverables: The proposed pricing model for the new product. Action points for the next steps, for example, obtaining additional information.

Post factum analysis: Once established, the pricing model can be tested in practice and adjusted based on the reception by the market.

Key design decisions: Using the Accion framework to guide the discussion. The framework is one of a few action-oriented approaches to pricing (Saltan and Smolander, 2019). Saltan and Smolander (2019) call for further practical evaluation of several pricing frameworks, including Accion. Moreover, the latest edition of the ISPMA-compatible study guide and handbook (Kittlaus, 2022) includes a description of the framework as a practical approach to pricing.

4.2.5. Legal review

Name: Legal review

Impacted elements of product strategy: Legal and IPR management

Tools: Checklist of legal considerations that must be reviewed.

Participants: Project manager, Legal department in a company (if available), Data Protection Officer, COO

Approach: Open discussion about 4 legal aspects that need to be considered. Separate meetings or discussions can be held for each of the 4 aspects. For each aspect, the following questions must be discussed:

1. Contracts. Who is responsible for formulating service contracts in the organization? What are the terms of the service level agreement? Does the organization have templates for such SLAs? Can any existing SLAs be reused?
2. IPR protection. How will the company protect the intellectual property rights related to the product? Does the company have patents or trademarks that apply to the product? Should the company obtain new trademarks or patents for the product?
3. Open source. What open-source components may be used when developing and running the software? What are the distribution licenses for those components? Are there any restrictions or caveats?

4. Data protection. Who is responsible for formulating the privacy policy within the organization? What are the terms of the policy? Does the organization have templates for such policies? Can any existing policies be reused?

Additional legal aspects may be necessary to consider depending on the context.

Preparation: The product manager identifies the right experts to have the legal discussions with. Executive management can be consulted if it is unclear who is responsible for contracts and legal aspects. If a dedicated legal department is available, the product manager looks for the appropriate legal experts to discuss the new product with. If there is no dedicated legal department, the product manager determines who oversaw these legal aspects in past projects.

Process:

1. Contracts.
 - 1.1. The product manager sets up a meeting with a person deemed responsible for formulating contracts, or the person with experience in making contracts within the organization.
 - 1.2. The product manager briefly presents the current product strategy to provide context for the discussion.
 - 1.3. The legal expert is asked to share whether the organization has templates for SaaS contracts and whether the contracts from any previous products can be reused or adapted. The overall terms of the agreement are discussed.
2. IPR protection.
 - 2.1. The product manager sets up a meeting with the person at the company who is knowledgeable about the current IPR protection processes.
 - 2.2. The product manager inquires what trademarks and patents are held by the organization. The product manager evaluates whether these apply to the current product. The product manager also consults the legal expert whether new patents or trademarks can or should be obtained for the new product.
3. Open source.

- 3.1. The product manager identifies the open source components used or intended to be used when developing the product.
 - 3.2. For each identified component, the product manager checks its distribution license.
 - 3.3. The product manager evaluates the implications of the license. A legal department can be consulted if available.
 - 3.4. The product manager approves or rejects the components based on the implications of their licenses. In case of rejection, the product manager discusses possible replacements with technical leadership.
4. Data protection.
 - 4.1. The product manager sets up a meeting with the person deemed responsible for formulating privacy policies and overall data protection. European companies may have a dedicated Data Protection Officer role.
 - 4.2. The product manager briefly presents the current product strategy to provide context for the discussion. The product manager highlights areas where personal data is collected or processed.
 - 4.3. The legal expert is asked to advise the product manager on the relevant legislation and actions that need to be taken to ensure data protection and legal compliance. This may vary based on the organization and country. Some of the points that may be raised:
 - 4.3.1. Can any existing company policies be reused for the new product?
 - 4.3.2. Should the product prompt the users to accept the cookie policy and the privacy policy when they use the product for the first time?
 - 4.3.3. What are the possible regional restrictions?
 - 4.4. The product manager takes notes and makes a list of actions to be taken next.

Produced deliverables: A list of action points on what documents need to be created. Links to templates that can be used or documents that can be adapted. List of brand names that need to be trademarked, or concepts that need patent applications. List of open source components or license types that are allowed or not allowed to be used.

Post factum analysis: The product manager documents the main decisions made for each aspect, and the actions that need to be taken next in the product strategy document.

Key design decisions:

1. Splitting this phase into several meetings, each with the stakeholder or stakeholders that have the most experience in the related area.
2. Relying on existing organizational knowledge of legal aspects as much as possible.

The last phase discusses performance and risk management aspects of a strategy.

4.2.6. KPI selection & risk analysis

Name: KPI Selection and Risk Analysis

Impacted elements of product strategy: Performance and Risk Management

Tools: Balanced Product Scorecard, “Digital Red Dot Game”

Participants: Representatives from executive management, Development, Sales and Marketing give feedback on the selected KPIs and the strategy overall.

Approach: The product manager selects KPIs guided by best practices. The product manager asks stakeholders to point out strategy elements they are unsure about.

Preparation: The product manager inquires whether the organization has a dedicated KPI team, or already keeps track of certain metrics in some way.

Process:

1. The product manager adds the business goals stated during the collaborative strategy workshop to the Balanced Product Scorecard.
2. For each business goal, the product manager sets a target.
3. For each business goal, the product manager describes how it will be measured.
4. The product manager selects metrics from the sample KPI list (Appendix 1), as well as metrics already used at their company (if available). The product manager follows the best practices in KPI selection (tying KPIs to business goals, ensuring every KPI is measurable, avoiding vanity metrics, and considering all 4 perspectives of the

Balanced Product Scorecard). Additional perspectives can be added to the scorecard if deemed necessary by the product manager or the KPI team at the organization.

5. For each KPI, the product manager describes how it will be measured.
6. The product manager adds the filled Balanced Product Scorecard to the product strategy document.
7. The product manager shares the product strategy document with representatives from executive management, Development, Sales, and Marketing. Ideally, this is done via an online text editor where the reviewers can see each other's comments. The product manager asks for feedback on the overall strategy. The product manager also asks the reviewers to mark three statements or aspects of the strategy that they are least confident about. This is what is referred to as the "Digital Red Dot Game". The concept of the Red Dot Game was explained at the end of Section 2.5.8.
8. The product manager analyses the feedback and makes changes to the selected KPIs and the overall strategy if necessary.
9. The product manager makes a list of risks based on the areas that were marked by reviewers as uncertain.

Produced deliverables: The Performance and Risk Management section of the product strategy document is written. KPIs for the four perspectives are specified, including how they will be measured. The list of risks for the product is compiled.

Post factum analysis: The product manager makes corrections to the KPIs based on feedback and compiles a list of risks. The product manager prioritizes the risks if it is possible to prioritize them based on the elements of the strategy marked as risky by reviewers.

Key design decisions:

1. Requesting feedback on the KPIs, the strategy overall, and the riskiest elements at the same time. This is the last phase of the process model. At this point, the product strategy document should be mature enough to be reviewed as a whole.
2. Suggesting the Balanced Product Scorecard as the tool for this phase and the sample KPIs for the Scorecard formulated by Pichler and included in Appendix 1. The

Scorecard and the sample KPIs provide a basis for holistic KPI selection. The process acknowledges companies may have their own categories and metrics, and does not prohibit incorporating them as well.

3. Collecting feedback asynchronously in form of comments on the product strategy document. Bavaro and McDowell (2021) write that collecting feedback asynchronously allows the product manager to receive everybody's opinions before formulating a response, or deciding on what needs to be changed.
4. Limiting risk management to risk identification and prioritization. Excluding the creation of contingency plans and establishing a continuous monitoring process at this point. This is primarily due to the time constraints of the research and is mentioned in the limitations in the Discussion chapter.

This concludes the description of the process model. In the next chapter, the model is demonstrated at the commissioner company.

5. Model demonstration: creating a strategy for Perception Cloud

In this chapter, the model is demonstrated. The process is followed to develop the product strategy for the productized SaaS version of the Perception system.

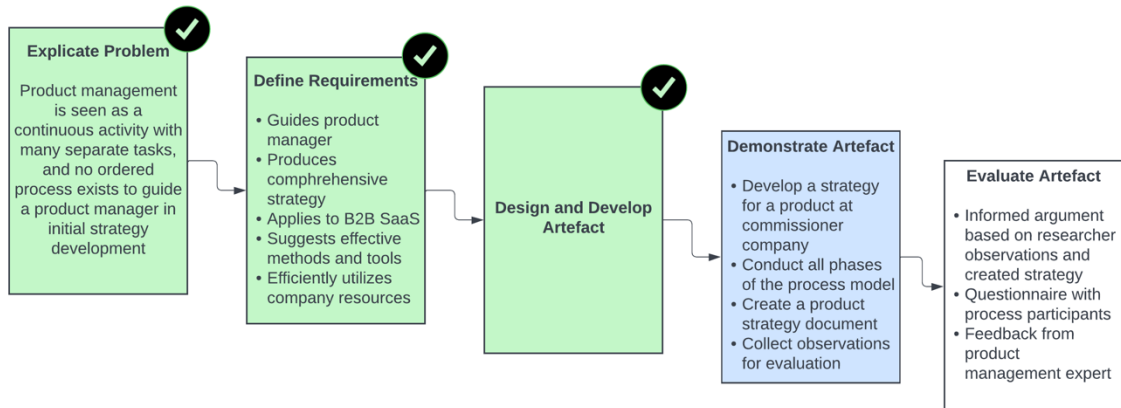


Figure 24. Design research process after designing the artefact

Figure 24 displays the current stage of the design research process. Having designed the process model, it is time to demonstrate it in a real-world scenario.

5.1. A brief history of the Perception system

The Perception system has undergone several major phases in its development. The first version of the system was offered as a standalone vehicle installation around 2016. The system was installed on a ruggedized computer and consisted of a client application and a server application referred to as the Perception Box. The system provided a simple user interface (UI). The system was designed to be installed in a CBRN reconnaissance vehicle and provide awareness to the operators inside the vehicle. No reach-back functionality to the command centre was implemented at this stage. CBRN detectors were connected to the Perception Box and the UI displayed the layout of the vehicle from the top, with the locations of each detector within the vehicle. The measurements and status of the detectors were displayed on the UI, and visual and audio alarms were triggered when device measurements bypassed certain configurable dangerous thresholds. The system also provided a map that displayed the location of the vehicle. At this point, Perception was an independent project, at stage 1 of the productization process by Artz et al. (2010).

The next major version of the Perception system was developed in 2017-2018. Still a standalone single-tenant installation, this version of the system was designed for a shelter case. Instead of a vehicle, multiple stationary devices were integrated into the software. Several Boxes with connected devices communicated to a central server. The user was able to manually configure the locations of stationary equipment on a map. The system was expanded with many features, including user management, device configuration options, scheduling and tracking of maintenance tasks, a library of response procedures in case of incidents, and simple automation rules that would trigger certain procedures in case of an alarm. Minimal chat functionality as well as SMS integration for sending out alerts to users was added. The codebase of the original project was partially used as the basis for the new project, but the codebases remained separate, and Perception was still at stage 1 of the productization process by Artz et al. (2010).

The company started a new vehicle project, with another shelter project on the horizon. It became apparent that splitting the codebase for each new project will not be sustainable long-term. Moreover, many features already developed for previous projects could be reused with enhancements and customizations for the new projects. This was the beginning of the so-called Perception 2.0 initiative. The idea was to share the same codebase for every project. For every project, the software could be adjusted using configuration files to fit unique customer needs. On the back end this proved easier, because a modular architecture was used there from the beginning. Certain modules can be added or removed based on the project. For the front end, however, a new solution was needed. A new client application was developed. A proprietary application model syntax was created. Using the application model, it became possible to modify the contents of the client application, by adding and removing panels, windows, and components. With this approach, a single client application could be used both for shelter and vehicle installations. Moreover, a visual editor for the application model was developed. A client could be launched in “designer mode”, where it was possible to drag and drop certain components onto the UI and adjust their properties.

The new client was presented at a company keynote, and during the demo a shelter-focused UI was rebuilt into a vehicle-focused UI within minutes. CI/CD pipelines were configured to build the client with different application models and settings depending on the project name provided as an environment variable. At this point, the product went through stage 3 of productization in the Artz et al. (2010) classification – product recognition – and entered

stage 4 – product platform. The amount of customized code was still quite large, but separate projects were built on top of a generic basis. The commissioner company identified the shared needs of separate customers to form this generic basis. Using the application model, derivative client applications could be created for different customers. But the technical implementation still had limitations.

The next big development was the creation of the Perception Go mobile app in late 2019. The application allows to pair portable CBRN detectors via Bluetooth. The application was developed within a new folder within the same code repository. The application did not reuse any of the existing code but was compatible with the main Perception system and sent measurement data via a message bus in real-time. The Perception 2.0 client was enhanced to display the location of the smartphone running the Perception Go app on a map, as well as the measurements of the CBRN detectors paired to the app. Perception Go was received enthusiastically in the CBRN industry, as it allowed connecting older devices that are perfectly capable but miss networking capabilities to a modern software system. The development of the Perception Go app did not impact the stage of productization in the Artz et al. (2010) classification.

The next major chapter for the Perception system was the development of a white-label web-based version for a customer organization in 2021-2022. During this project, a web-based UI was developed for the first time. Previously only desktop and mobile applications were made for Perception. During the project, the commissioner company adopted two open-source components – Keycloak and Grafana – for user management and reporting respectively. The project served as a learning experience for the upcoming Perception Cloud, including the development of new features and a better understanding of customer and user needs. The Perception Go application was also customized and advanced for the project. At this point, however, the software did not support multitenancy or composability, as it was developed for a single customer with a fixed list of requirements. Moreover, the customer required access to the codebase, and a separate code repository was made for the project, using previous projects as a basis. In this project, it was not possible to bring the application model features from the desktop client to the web application. As a result, at the beginning of Perception Cloud development, there is no product platform for projects that require web-based user interfaces.

In summary, at the time of this writing, the desktop version of Perception is in stage 4 of productization (Product Platform), whereas the yet-to-be-developed Perception Cloud product is conceptually in stage 3 (Product Recognition).

5.2. Strategy design activities with key stakeholders

This section describes how each phase of the process model was implemented in practice. The author refers to himself as “the product manager” when describing these phases.

5.2.1. Collaborative strategy workshop

The first step of the proposed process model is the collaborative strategy workshop. The workshop was conducted at the office of the commissioner company on April 27th, 2022. The COO and CTO of the company, as well as a project manager and a sales representative were present in the meeting with the product manager. The discussion was structured around the Product Vision Board tool by Pichler. Figure 25 shows the functional areas of the commissioner company whose representatives participated in the phase.

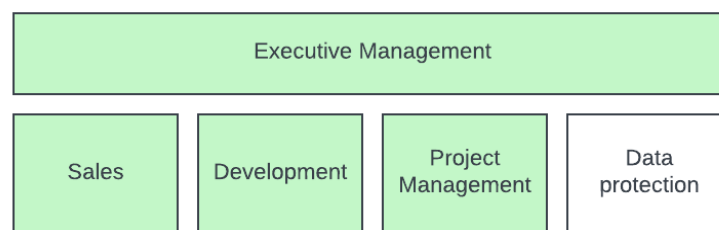


Figure 25. Functional areas of commissioner company involved in the collaborative strategy workshop

While the first section of the Product Vision Board is the product vision, the participants found it challenging to formulate the vision as the first step during the meeting. It was decided to go through all the other sections at first and see the vision that emerges. In general, it was made clear that the main goal for the product is to simplify the installation process of the Perception system and to unlock new revenue streams. The name Perception Cloud was agreed upon for the product.

Target group. The on-premise Perception system is currently offered to civil defence and military organizations. For the military, a SaaS offering is currently out of question. In the

future, they may become open to private cloud installations. For Perception Cloud, the identified target groups were civil defence, emergency services, first responders, customs, and border control. Industrial business is another possible group. To focus the discussion, first responders were selected as the first target group for the initial version of Perception Cloud. Specifically, a firefighting organization was discussed.

Needs. The target group has many CBRN detectors that take measurements in the field. The data from these detectors need to be exported manually one by one, the data is not grouped and not summarized automatically. During a CBRN incident, it is difficult to establish a complete operational picture to enable decision-making. Multiple parties are involved in handling a CBRN incident, including the police, ambulances, and fire department, as well as government agencies and ministries. First responders would need a quick way to share information with all the parties that need it. Knowledge from the field needs to be quickly delivered to decision-makers. For the customers of on-premise Perception systems, maintaining the IT infrastructure is an additional challenge that would be eliminated by the SaaS offering.

Product. The core features of the product discussed were data gathering of CBRN measurements, reporting tools displaying charts with measurements from multiple devices over time, and a map displaying the locations of detectors. Compatibility with a wide variety of CBRN detection equipment was agreed upon as a must. The product should provide open APIs to allow integrators to make their systems compatible with Perception Cloud. Additional work can be done to integrate the proprietary systems of first responders into the Perception Cloud. First responder organizations have their own alarm systems and vehicle networks which could be supported by Perception Cloud. It was agreed that the product should be modular, with extra features available at extra cost.

Business goals. Perception cloud can become a new revenue source for the commissioner company. The installation process would be faster and cheaper compared to the on-premise Perception solution. Moreover, Perception Cloud can be used to market and sell another product - the Perception Go app.

Competitors. Several competitors were identified for the system.

- CBRN responder by the US government (see CBRNResponder, 2022)

- Honeywell Safety Suite SaaS. Primarily targets the industrial sector but can also be used by first responders. The website provides a breakdown of features that can be used in the strategy canvas (see Honeywell, 2022b).
- Another Honeywell product is ConneXt Gas Monitoring, but limited information seems available about its features (see Honeywell, 2022a)
- Companies that offer on-premise CBRN SAS were also mentioned as possible competitors for the Perception Cloud product.

Revenue streams. With SaaS this is subscriptions. The pricing structure was not discussed in detail in this case. It was mentioned that extra features may be available at extra cost, and it could be possible to have subscription tiers that allow connecting an increasing number of CBRN detectors to the system.

Cost factors. The development effort was mentioned as the largest cost factor. One possible paid third-party software component for reporting was discussed. The possibility to outsource customer support was mentioned, especially given the intended global scope of the product.

Channels. Some of the channels already used to market and sell the on-premise version of the Perception system can be utilized to market and sell Perception Cloud as well. This includes expos, magazine ads, and private demonstrations. The product can also be promoted via the partner device manufacturers of the commissioner company. Multiple partner companies were mentioned.

Table 6. Problem and position statement for Perception Cloud

Problem statement	
the problem of	lack of complete vision of the operational picture during a CBRN incident
affects	civil defence, emergency services and first responders
the impact of which is	increased delay in response to incidents and increased harm to the wellbeing and lives of victims
a successful solution	unifies data from multiple types of CBRN detectors to provide a comprehensive operational picture to decision makers, reducing the time necessary to make informed decisions that save lives.

Position statement	
for	civil defence members, emergency services, first responders, as well as border control and customs officials
who	respond to a CBRN incident to reduce the hazard and avoid harm to the population in the area of incident
the	Perception Cloud solution
that	provides a centralized interface to view the collected measurements of a variety of CBRN detection devices
unlike	the current approach of manually collecting and correlating data from multiple CBRN detection devices
our product	reduces response time to CBRN incidents by supporting informed decision-making based on a comprehensive operational picture.

Based on the discussion, a problem and position statement was developed. It is displayed in Table 6.

5.2.2. Product definition refinement

The second step of the proposed process model refines the initial understanding and definition of the product by comparing it to competing offerings from the industry. In preparation for this step, the product manager has evaluated the competitor products in the industry. The two products evaluated were the aforementioned Honeywell Safety Suite and the CBRNResponder. It is worth noting that CBRNResponder is currently not a commercial product and is only offered to US-based organizations involved in chemical, biological, radiological or nuclear preparedness, training, response, recovery or other elements of technical hazard engagement. The Honeywell Safety Suite is marketed to first responders (the primary customer segment of Perception Cloud) as well as industrial plant workers (prospective future customer segment of Perception Cloud).

To evaluate the Honeywell Safety Suite, the product sheet from the official website was used. A section of the product sheet is presented in Figure 26. While lacking detailed descriptions of the features, it was sufficient to pinpoint the key functional areas.

Module	Category	Feature	Deluxe	Premium
Device Management	Single-Device Operations	Device & Sensor Configuration	●	●
		Firmware Updates	●	●
		Download, Store, View & Export Data	●	●
		Event, Calibration & Bump Test Reporting	●	●
		User Management – Setup Users & Assign Devices	●	●
	Multi-Device Operations	Multi- Device Configuration Templates	●	●
		Multi-Device Download, View & Export Data	●	●
		Multi- Device Event, Calibration & Bump Test Reporting	●	●
	Customization	User Defined Fields	●	●
		Reports Customization	●	●
	Advanced Features	Historical Data Dashboard	●	●
		Bump Test & Calibration Forecast	●	●
		Worker Device Assignment Kiosk	●	●
Report Repository		●	●	

Figure 26 Section of Safety Suite product sheet (Honeywell, 2022b)

To evaluate the CBRNResponder software, the product manager viewed a nationwide introductory training seminar for the software that took place on April 22, 2021. The webinar is freely available on YouTube (see CBRNResponder, 2021) and provides a detailed insight into what the product offers and how the customers use it. During the seminar, first, the features of the software are introduced, and then multiple emergency response organizations throughout the US describe how they use the software, what they find valuable about it, and what are their pain points and advice for adoption.

Finally, the product manager considered the current feature set of the Perception system and compiled the list of features and categories on which the new Perception Cloud software could compete with the industry offering. In this case, the product manager was well familiar with the feature offering of the customer-tailored Perception system and did not require consulting a product sheet. But if a product manager is not closely familiar with the current software, consulting a product expert or at least the marketing, proposal or training materials for the current software is advisable. The product manager created a blank Strategy Canvas using a free online tool available from Visual Paradigm. Figure 27 presents the Strategy Canvas prepared by the product manager as the entry point for the product definition refinement meeting.

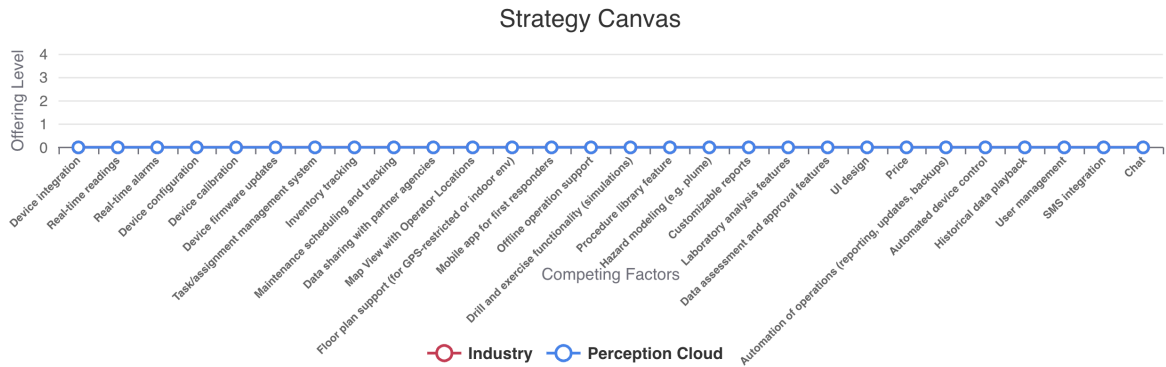


Figure 27 Strategy Canvas with unweighted competing categories

The product definition refinement meeting was conducted at the office of the commissioner company on May 23rd, 2022. A project manager and a sales representative were present in the meeting with the product manager. Figure 28 shows the functional areas of the commissioner company whose representatives participated in the phase. The discussion was structured around the Strategy Canvas tool. At the beginning of the meeting, the product manager introduced the tool. The Strategy Canvas for the first iPhone was presented as an example.



Figure 28. Functional areas of commissioner company involved in product definition refinement

The product manager also reminded the participants what competitor products were used as the industry representatives when the categories were compiled. After this point, the participants started going through each category. The offering level in the industry and the desired level for Perception Cloud were estimated. After checking several of the categories, a suggestion was made to add a third line to the canvas – the current Perception system. This allowed to compare the existing Perception software to the industry offering and highlight which areas of the existing software need to be enhanced or created for the new Perception Cloud. During the meeting, several categories were eliminated from the canvas, and several

new ones were added. The Strategy Canvas produced at the end of the meeting is presented in the figure below. Figure 29 shows the Canvas the way it looked at the end of the meeting.

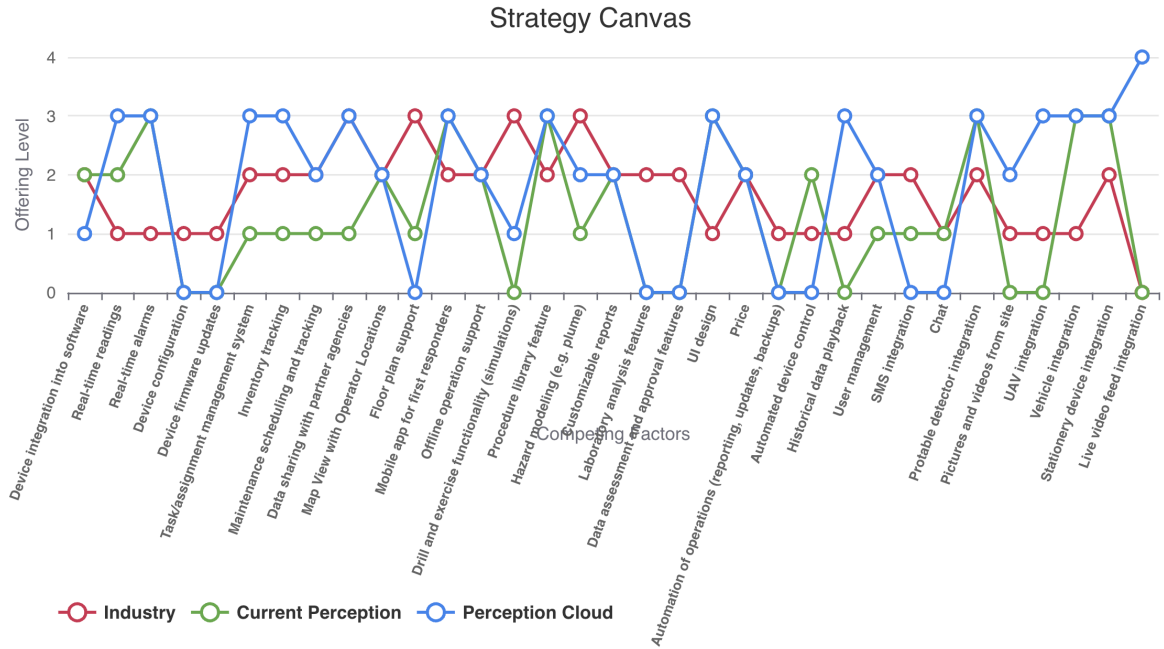


Figure 29. Strategy Canvas for Perception Cloud at the end of the meeting

After the meeting, the product manager sorted the canvas by the estimate values given for Perception Cloud, from lowest to highest. The result is displayed in Figure 30.

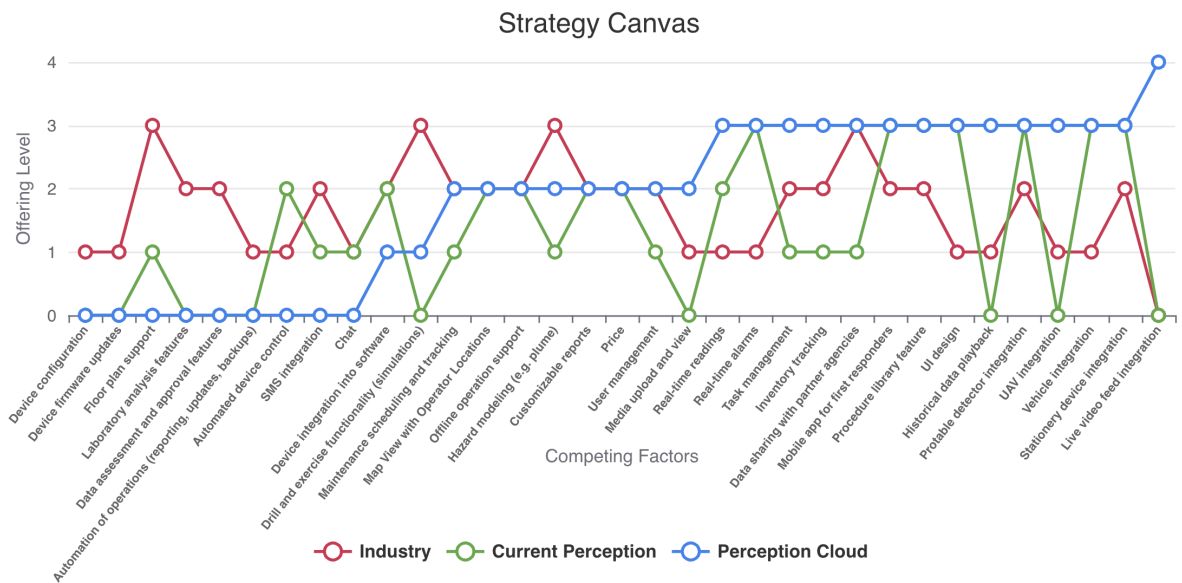


Figure 30. Strategy Canvas for Perception Cloud combined with ERRC

Figure 31 shows the same canvas with the “Current Perception” line hidden, to focus specifically on the new Perception Cloud product.

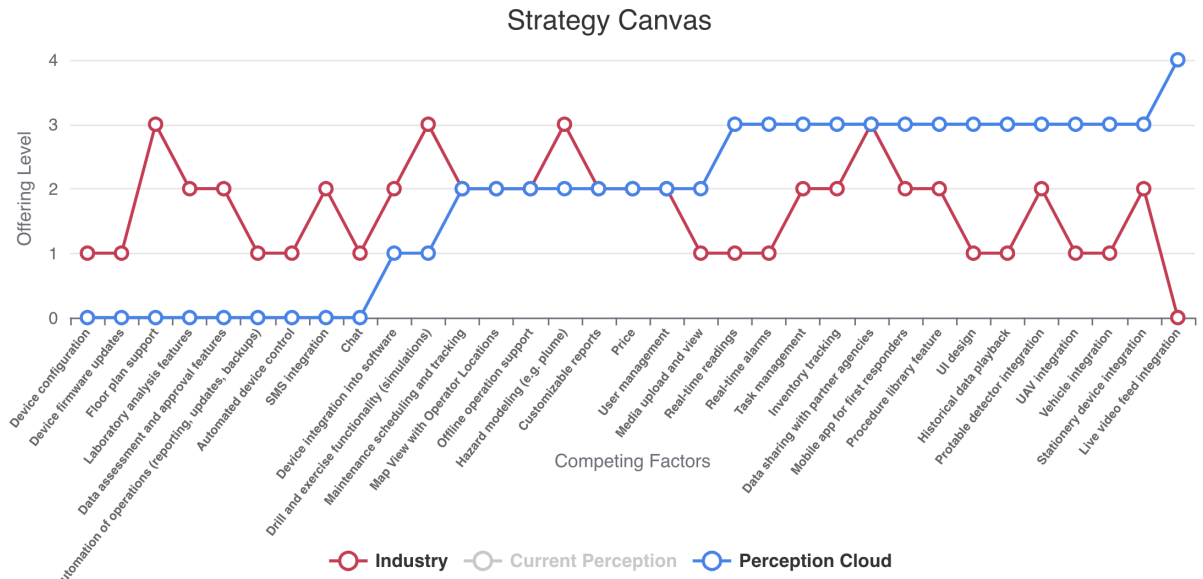


Figure 31. Strategy Canvas for Perception Cloud combined with ERRC, and current system value curve hidden

The canvas illustrates which features it was decided the Perception Cloud product should not offer (at least in its initial version), which features are acceptable to offer at a reduced capacity to the competition, which categories aim to match the competition and which categories aim to go beyond the competition. One category was identified that is currently not offered by the competition at all, indicating a blue ocean opportunity.

As established during the collaborative strategy workshop, the core purpose of the Perception Cloud product is to quickly offer a comprehensive operational picture to enable informed decision-making. In light of that, features related to real-time measurement communication and displaying were prioritized over features concerning the post-factum analysis of the data.

Table 7. Competing factors of Perception Cloud grouped by categories and sorted by level of offering

Category	Feature	Industry	Current Perception	Perception Cloud
Real-time	Live video feed integration	0	0	4
	Real-time readings	1	2	3
	Real-time alarms	1	3	3

	Procedure library feature	2	3	3
	Task management	2	1	3
	Data sharing with partner agencies	3	1	3
	Map View with Operator Locations	2	2	2
	Hazard modelling (e.g. plume)	3	1	2
	Media upload and view	1	0	2
	Floor plan support	3	1	0
Device integration	Mobile app for first responders	2	3	3
	Portable detector integration	2	3	3
	UAV integration	1	0	3
	Vehicle integration	1	3	3
	Stationery device integration	2	3	3
	Offline operation support	2	2	2
	Device integration into software (depth of integration)	2	2	1
Inventory management	Inventory tracking	2	1	3
	Maintenance scheduling and tracking	2	1	2
	Device configuration	1	0	0
	Device firmware updates	1	0	0
Post-factum analysis	Historical data playback	1	0	3
	Customizable reports	2	2	2
	Laboratory analysis features	2	0	0
	Data assessment and approval features	2	0	0
Competitive factors	UI design	1	3	3
	Price	2	2	2
Administration features	User management	2	1	2
Automation	Automation of operations (reporting, updates, backups)	1	0	0
	Automated device control	1	2	0
Preparation	Drill and exercise functionality (simulations)	3	0	1
Communication	SMS integration	2	1	0
	Chat	1	1	0

Table 7 groups related competing factors into broader categories. In each category, the competing factors are sorted by the level of their offering from highest to lowest. The background colour of the rows within a category is made darker based on the level of offering within Perception Cloud – from white background for offering at level 0 to dark grey for offering at level 4.

5.2.3. Architecture design

The first meeting of this phase took place on June 1st, 2022, at the office of the commissioner company. The participants included the CTO and a software engineer. Figure 32 shows the functional areas of the commissioner company whose representatives participated in the phase. In accordance with the proposed process model, the first meeting was a free-form discussion around a set of questions about the offering architecture, including tailorability architecture.



Figure 32. Functional areas of commissioner company involved in architecture design

What is the defining technology for the software product? What technology enables our competitive edge and market differentiation over time?

It was agreed that the existing mobile application Perception Go together with the backend services processing data from Perception Go is the technology that currently makes the commissioner company stand out in the CBRN software industry. The Perception Go app allows to pair portable detectors via Bluetooth and send observations to the backend system in real-time. Competitor systems require users to stop periodically to take measurements with their portable equipment and upload those measurements to the backend. This method of data collection is slower and may be especially challenging for users wearing protective gear, such as hazmat suits. The Perception Go app continuously transmits the measurements of the paired detectors to the backend and mostly does not require human interaction after the detector is paired. The application can run in the background with the screen turned off,

allowing operators to relay information to command centres in near-real-time, without even taking the phones out of their pockets. The proprietary backend services of Perception process and store incoming measurements efficiently.

What is the offering architecture? Meaning what are the separately priced components of the product?

It was discussed that the Perception Go application is clearly a standalone component that can relay data to the Perception Cloud system, as well as the standalone Perception system installations. But since Perception Go is the defining technology that makes the Perception Cloud attractive, it does not make much sense to sell the application separately, at least in the first iterations of the product. Perception Go, however, is only designed to integrate portable detectors into the system. As discussed during product definition refinement, the Perception Cloud shall also support integrations of stationary devices, vehicles, and UAVs. Once all those options become available in Perception Cloud, it could be possible to make Perception Go one of the optional separate parts of the offering. To give an example, a customer that only possesses stationary equipment (e.g., radiation gates) could opt out of the Perception Go part of the offering.

What is the tailorability architecture?

To answer this question the supplementary questions regarding the offering were discussed.

How configurable do we want the software to be? What parameters are configurable?

As described in the section “Brief history of the Perception system”, the existing Perception software is significantly configurable. It was agreed that the commissioner company would like to support some degree of configurability in Perception Cloud too. An organization with access to Perception Cloud would be able to configure, at minimum, the alarm limits for their detection equipment. In cases of integrating stationary equipment, it would be possible to configure the positions of each device on a map.

How composable do we want the software to be? What components can be added or removed?

It was agreed that composability should be a major focus for Perception Cloud. As discussed in the “Background” chapter, SaaS products are typically not customizable, but in the B2B market customers expect the products to be tailored to their business needs. A solution for

this is tailorability, through composability and configurability. It was discussed that Perception Cloud shall support multiple plugins that will be priced separately from the core offering. This way it would be possible to provide value to a more diverse set of customer segments while preventing the software from being overly tailored to a specific group.

It was discussed that the main part of the offering shall include the map where real-time measurements can be displayed, the Perception Go app that allows sending real-time measurements into the system, and an API that other stationary equipment can send data to. The possible plugins for the software include the playback of historical data (replaying events for post-factum analysis), reporting functionality, as well as various hazard propagation features, such as plume modelling.

How customizable do we want the software to be? Are we willing to modify the software code for specific customers?

As discussed above, non-generalizable customer needs can be addressed by developing customer-specific plugins, without modifying existing software code. The commissioner company would evaluate on a case-by-case basis whether investment into developing a customer-specific plugin is worthwhile. The commissioner company would also seek to generalize customer-specific plugins as much as possible so that they can be reused with future customers.

What is our desired place in a software ecosystem? What organizations could we partner with? What external systems could we integrate? Could we provide a way for third parties to integrate into our software?

In the meeting, it was discussed that the commissioner company would be interested in growing its role within a software ecosystem. At the moment the commissioner company is a niche player, with the potential of becoming a keystone player. One ecosystem is that of detection device manufacturers. The commissioner company has a set of trusted hardware partners and is the official reseller in Finland for some of the manufacturers. Manufacturers are interested in integrating their devices into the commissioner company software. Bidirectional network effects can be observed in the relationship between the commissioner company and device manufacturers. If the Perception Cloud product becomes successful, customers may wish to purchase devices supported by Perception Cloud, leading to increased sales for the manufacturers. On the other hand, if Perception Cloud supports many

types of detection devices, this makes the platform more attractive and leads to increased sales for the commissioner company.

The commissioner company could make it easier for willing manufacturers to integrate devices into Perception Cloud by providing an open API. It was noted that Perception Cloud will support different types of device integrations. Portable detectors will be paired to the Perception Go mobile app. Stationary detectors will be connected to a Perception Box— a proprietary computer that can be supplied by the commissioner company. The Box would forward device data to Perception Cloud. Finally, an open API can be provided as the third way to send data to Perception Cloud. Willing manufacturers can develop their own software compatible with the API.

Another point of discussion was the plugin ecosystem. The possibility of open-sourcing plugin development for Perception Cloud was discussed. This way willing parties would be able to create their own custom plugins. To enable this ecosystem, separate APIs are needed. Plugin developers need to have access to APIs to get data from the core Perception Cloud system and to send the data back into the system for the plugins to work. It was agreed that this open-source plugin ecosystem will not be implemented in the early versions of Perception Cloud.

At the end of the meeting, the overall architecture and technical constraints were discussed. Several specific technologies were agreed upon for Perception Cloud. Two open-source components were also identified – the Keycloak software for user management and the Grafana software for configurable reporting. All these choices were dictated by the existing Perception system. The consensus was to reuse as much of the existing Perception software as possible, to reduce development time and costs.

Finally, it was mentioned that certain customers of Perception Cloud may refuse to share the same servers with other organizations. Such organizations have stricter data protection requirements. The Perception Cloud system shall provide the capability to allocate dedicated servers to customers that request it. A dedicated server could come at an extra cost.

In accordance with the process model, the meeting adjourned after discussing the offering architecture, the software ecosystem, and the overall technology stack. The business architecture discussion was scheduled for the following week. Based on the discussion, the product manager prepared a draft of a domain model for Perception Cloud and identified at

least one new business process required to support the Perception Cloud product. The product manager prepared a visual representation of the process using BPMN (Business Process Model and Notation). The draft of the domain model prepared by the product manager before the second meeting of the “Architecture design” phase is presented in Figure 33.

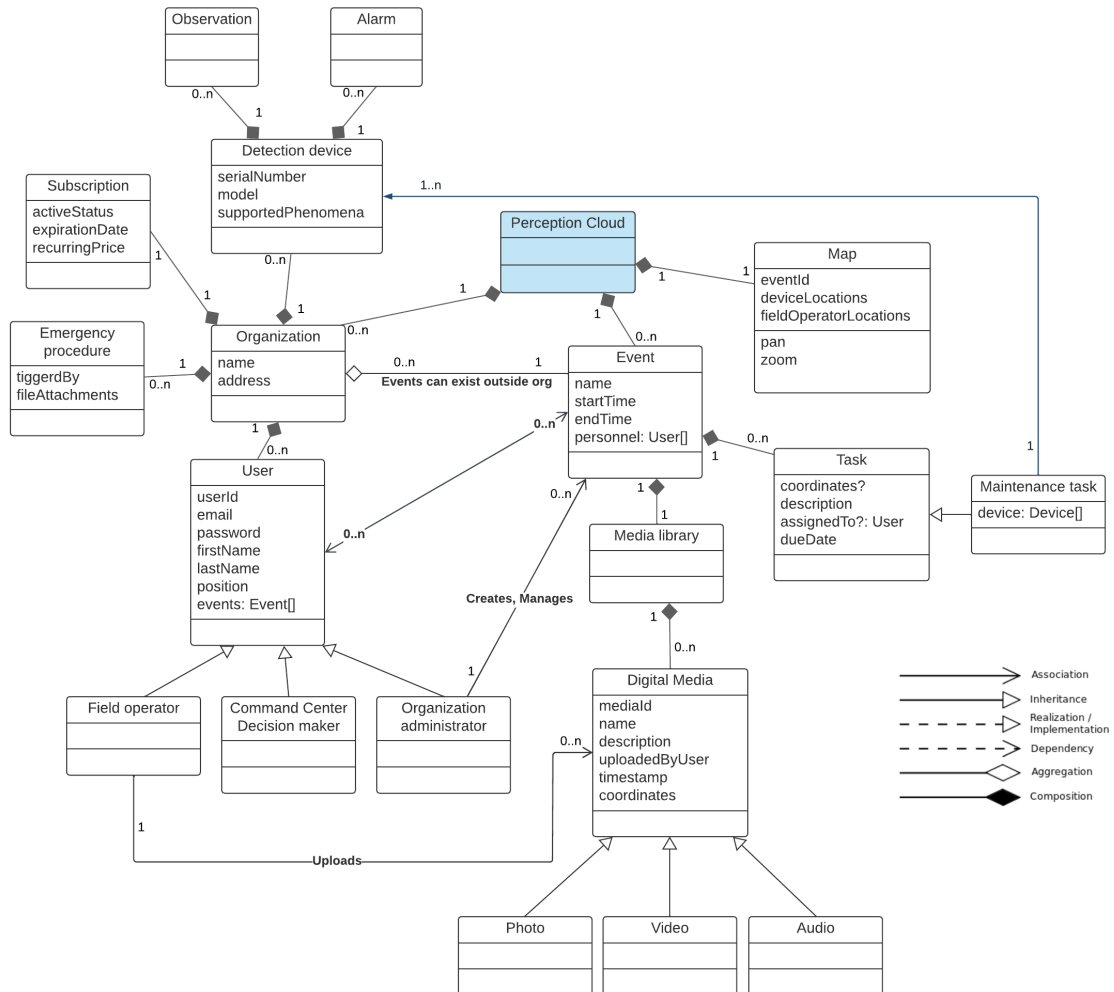


Figure 33. First draft of the domain model for Perception Cloud

Several key entities were identified, including:

- **Organization.** Organization has users of different types, and a fleet of detection equipment. Maintenance operations need to be performed on the equipment occasionally.

- Event (e.g., a CBRN incident). An Event has a Media Library where photos and videos of the incident can be uploaded by users. An Organization can create Tasks that need to be carried out during an Event.

The new business process identified by the product manager after the discussion was “adding a new customer to the Perception Cloud”. While the commissioner company has experience setting up the Perception system on customer premises, the commissioner company does not yet have an established process by which new organizations can be added to a multi-tenant cloud installation. The process model using the Business Process Model and Notation (BPMN) was prepared, using the Bizagi modeller software. The model is presented in Figure 35. Figure 34 demonstrates the subprocess of the main process – the “Development effort estimation”.

The second meeting of the “Architecture Design” phase took place at the office of the commissioner company on June 7th, 2022. The participants were the same as in the previous meeting – the CTO and a software engineer at the commissioner company. In the meeting, the product manager presented the domain model and the process model. Both models were opened in their respective editor software, and the product manager made changes and additions to the models during the meeting based on the comments.

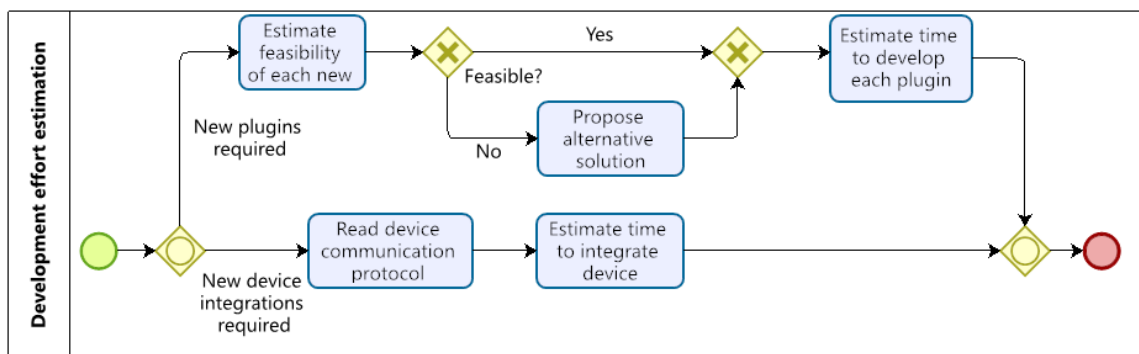


Figure 34. Development effort estimation subprocess

Overall, both models were accepted positively, as they summed up and built upon the preceding discussions. For the domain model, multiple additions were made. The second draft of the domain model, reflecting the changes made during the workshop, is presented in Figure 36.

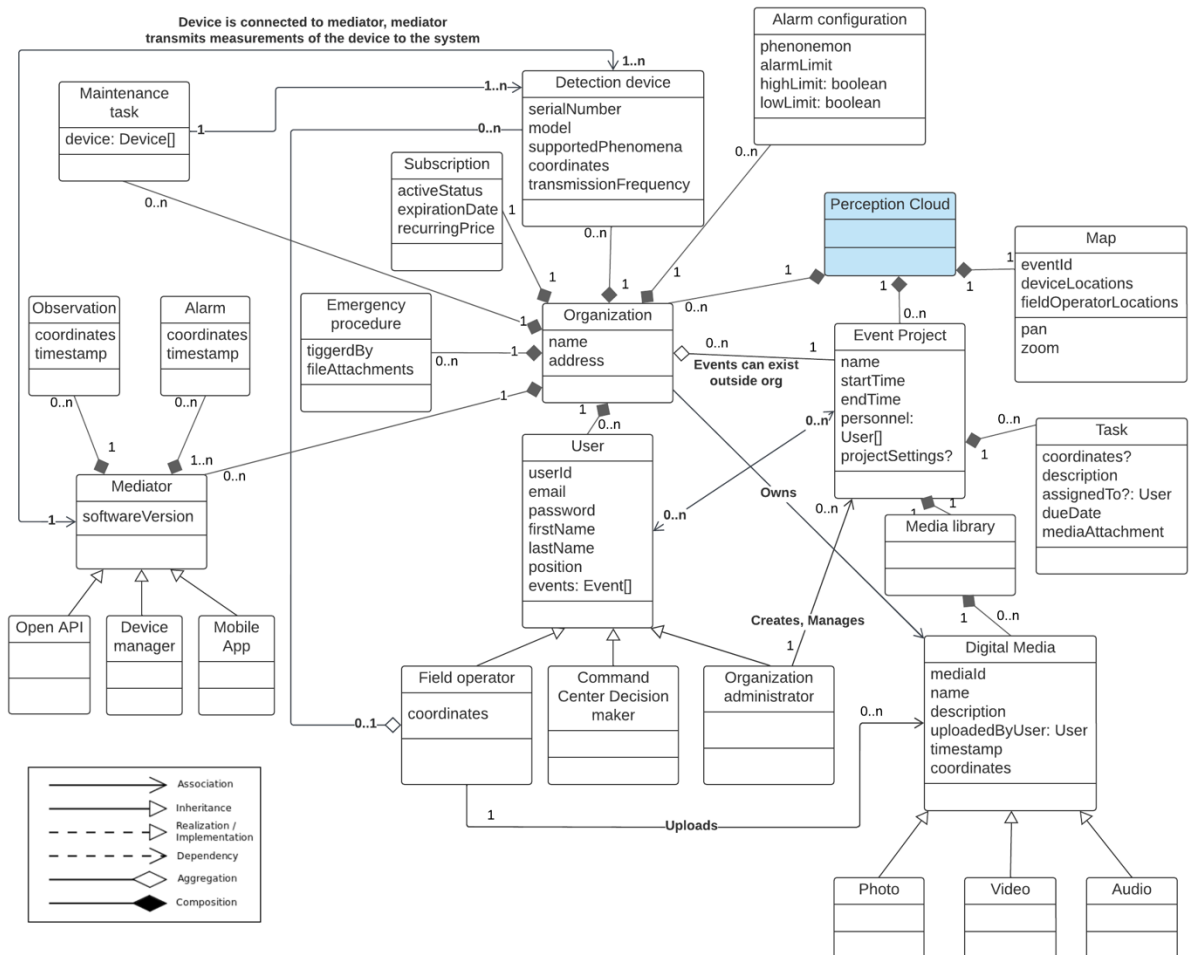


Figure 36 Second draft of the domain model for Perception Cloud

The following changes were made to the domain model as the result of the workshop meeting:

1. The relationship between Device and Alarm entities was changed from “one-to-many” to “many-to-many”. This is to reflect that the system can be configured to trigger certain alarms based on concurrent measurements from two or more devices. For example, the system can be configured to trigger alarms only when dangerous radiation values are detected by two devices at the same time.

2. The Event entity was renamed into “Event Project” for further clarity. The term “Event” has other meanings in programming.
3. Ownership of digital media was discussed. Event Projects can be accessed by multiple Organizations, and Users of each Organization can upload Digital Media to the Media Library associated with the Event Project. It was agreed that each uploaded Digital Media is owned by the Organization whose member uploaded the media.
4. Inheritance from Task to Maintenance Task was removed. Maintenance Task was changed to a separate entity. This was motivated by the fact that Tasks are created within Event Projects, whereas Maintenance Tasks are created within an Organization and are unrelated to specific Event Projects.
5. It was discussed whether Location needs to be added as a separate entity. Instead of a separate entity, a “coordinates” property was added to the Device, User, Observation and Alarm entities.
6. The “Alarm configuration” entity was added.
7. The “Mediator” entity was added. A Mediator provides data from detection equipment into the system. Three types of a Mediator were outlined: Mobile App (the Perception Go app), Device Manager, and Open API. A Mediator has a “software version” property. Observation and Alarm entities used to have a composition relationship with the Device entity. This was changed so that the Observation and Alarm entities have a composition relationship with the Mediator. It is the Mediator that forwards the observations from devices to Perception Cloud and raises Alarms when dangerous limits are exceeded.

For the business process model, no major changes were suggested during the meeting. One point was made that when a new device needs to be integrated, the commissioner company needs to estimate the cost of the device and where the device can be obtained from. The commissioner company needs access to the new device to work on its integration.

5.2.4. Financial discussion

In preparation for this phase, the product manager listed the possible value metrics for Perception Cloud based on previous discussions. The list included the following items:

- Number of CBRN detectors
- Supported types (models) of CBRN detectors
- Number of users
- Number of features (there are multiple plugins that add features to the software)
- Capability to allocate dedicated servers to customers that request it. A dedicated server would come at an extra cost, and a cheaper tier of Perception Cloud would share hardware resources between multiple organizations
- Used storage/server resources. Organizations can upload photos and videos to the system

The financial discussion was conducted at the office of the commissioner company on July 7th, 2022. A project manager and a sales representative were present in the meeting with the product manager. Executive management provided their feedback later on. Figure 37 shows the functional areas of the commissioner company whose representatives participated in the phase. The discussion was structured around the SaaS pricing framework by Accion. At the beginning of the meeting, the product manager reminded the participants about the key outcomes of the previous phases of the process model. Specifically, the product manager briefly presented the problem and position statement from the collaborative strategy workshop phase, the prioritized list of features from the product definition refinement phase, and the domain and business process models from the architecture design phase.



Figure 37. Functional areas of commissioner company involved in financial discussion

After this, the Accion pricing tool was introduced. Accion provides a slide deck explaining the framework and SaaS pricing best practices. During the meeting, the product manager used the slides to guide the discussion. The participants found the Accion example of pricing a Picasso painting using the cost-based, competitor-based, and value-based approaches insightful. The four steps of the pricing frameworks were followed.

1. Defining the upper bound

- a. *The uniqueness of the product.* It was agreed that the Perception Cloud product is unique (as seen earlier in the process, only a few indirect competitors were identified). Because the product is unique, the commissioner company can act as a “price maker”. Competing with other companies based on price is not necessary at this stage.
- b. *Contributing to customers’ bottom line.* The Perception Cloud product reduces labour costs for customer organizations. Making reports, aggregating detector data, and data sharing are largely automated using the software. The software also reduces the risk of errors compared to manual reporting and data aggregation. Strictly speaking, the software does not increase the revenue of the customer organizations. But the software aims to prevent harm to the priceless well-being and lives of people in areas of CBRN incidents.
- c. *Customer switching costs.* The Perception Cloud software aims to provide low switching costs to customers. The sales team would like to emphasize the ease of adoption of the software. Perception Cloud supports many types of detection equipment and offers customers to integrate their existing hardware into the system. Customers can gain the benefits of data aggregation, connectivity, and fast response times without having to purchase new detection hardware to be compatible.
- d. *Customer segments and their needs.* This has been established in previous phases of the process model and was not discussed much further. It was agreed that at the moment the customer segment is sufficiently narrow (first responders, emergency services, civil defence) and there is no need to break down the pricing structure by customer segment. In the future, when industrial organizations are targeted, additional segmentation can be added.

e. *How price-sensitive are the customers?* This may depend on the region of the customer. The product is a security system, and Accion mentions “reacting out of fear” as a factor that makes customers less price sensitive. The product addresses strategic anxiety, and customers may prefer demonstrating that they are taking it seriously by purchasing an expensive product (Accion, 2015). On the other hand, customers may be governmental organizations with stricter supplier oversight and budgetary constraints, which increases price sensitivity.

The participants attempted to follow the Accion upper bound calculation process (Table 8), but the result was inconclusive. The positive or negative direction of the “Pricing implication” was easier to estimate for each question than the actual number.

Table 8. Determining the upper bound for Perception Cloud according to the Accion framework process

Question	Answer	Pricing implication	Approximate upper bound
How does this product reduce customer costs?	* Reduces labour costs on data aggregation, reporting, and data sharing * Decreases risk of data entry error	+10000	10000
How does this product increase customer revenue?	Saving a human life is priceless. Further comparison with the insurance industry could be useful. Overall, the product is more about reducing costs and making life easier.	Neutral	10000
How unique is this product?	The product is unique, the sector is specific and direct competitors were difficult to identify.	+5000	15000
What type of switching costs do these customers face?	The switching costs are primarily on training to use the new software system.	-3000	12000
Do customers understand this product's value?	Potential customers show interest during demos of similar systems, see the benefit in data aggregation and comprehensive picture	+1000	13000
What are major customer segments?	First responders, emergency services, civil defence	Neutral	13000
How price sensitive are customers?	Somewhat price-sensitive as non-profit or government organizations, but willing to invest in security	Neutral	13000

2. Defining the lower bound

It was agreed that accounting data can be used to establish the cost baseline. As mentioned in the “Brief history of the Perception system” section, the commissioner company recently completed a customer project – a white-label version of the Perception system for a customer, with a web application as the front end. Accounting data can be consulted for that project to establish the expected monthly development cost.

3. Identifying reasons to charge less than the maximum value

As mentioned above, competing with other companies based on price was deemed unnecessary at this stage. It is possible to experiment with discounting the software a little when customers sign up for a full year instead of a month-to-month subscription.

4. Structuring the pricing model as a compromise between the upper and lower bound

Freemium, consumption-based, and tiered pricing were introduced by the product manager as the options for the SaaS pricing structure.

Several freemium elements were proposed. It was discussed that the Perception Go mobile app can be offered free of charge via mobile app marketplaces. The app can be offered with limited functionality free of charge. This would help advertise the paid Perception Cloud and generate sales leads for every customer that registers through the free app. Free trials for the Perception Cloud with limited features were discussed as a good way to encourage potential customers to sign up. A paid trial was also suggested for customers willing to unlock more features during the trial period.

Consumption-based pricing was deemed poorly suitable for the product. The product is a security system that is most actively used during and after the hopefully rare CBRN incidents. Tying the pricing to rare incidents would not be financially wise.

Regarding the tiered pricing, it was agreed that for the time being the size of the customer organization will be the main tier differentiator. As mentioned above, the customer segment for the first version of Perception is quite homogenous. Later on, industrial organizations can be added as a new segment, with possible separate pricing tiers.

The Accion slide deck ends with examples of pricing pages of various SaaS companies. The pricing pages are annotated, with pricing structures, value metrics and customer segments pointed out. The example pricing pages elicited more ideas from the participants of the meeting. After the meeting, the product manager created a “draft pricing page” for the Perception Cloud offering based on all the discussions so far. The draft pricing page is presented in Figure 38. The pricing page was shared with several stakeholders for further feedback, with the caveat that the actual prices listed on the draft are only placeholders which can be freely increased or reduced.

Essential	Professional	Premium	Premium+
<p>50€ a month (600€ billed annually)</p> <p>OR 70€ month-to-month</p>	<p>150€ a month (1800€ billed annually)</p> <p>OR 180€ month-to-month</p>	<p>300€ a month (3600€ billed annually)</p> <p>OR 400€ month-to-month</p>	<p>Need more devices or features, or a more flexible plan?</p> <p>Contact us</p>
No training fee	500€ training fee	1000€ training fee	
<ul style="list-style-type: none"> Up to 10 user accounts Up to 3 different detector types supported Up to 2 plugins included in subscription 	<ul style="list-style-type: none"> Up to 50 user accounts Up to 15 different detector types supported Up to 5 plugins included in subscription 	<ul style="list-style-type: none"> Over 50 user accounts Unlimited different supported detector types allowed All plugins included Possible dedicated server for your organization 	<ul style="list-style-type: none"> We can integrate your device to be supported by Perception Cloud We can implement custom features to fit your needs
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px 10px; background-color: #c6e0b4;">TRY</div> <div style="border: 1px solid black; padding: 2px 10px; background-color: #c6e0b4;">BUY</div> </div> <p>Free 30-day trial</p>	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px 10px; background-color: #c6e0b4;">TRY</div> <div style="border: 1px solid black; padding: 2px 10px; background-color: #c6e0b4;">BUY</div> </div> <p>Free 30-day trial</p>	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px 10px; background-color: #c6e0b4;">TRY</div> <div style="border: 1px solid black; padding: 2px 10px; background-color: #c6e0b4;">BUY</div> </div> <p>Free 30-day trial</p>	<div style="border: 1px solid black; padding: 2px 10px; background-color: #c6e0b4; text-align: center;">CONTACT US</div>
Included in all versions			
<ol style="list-style-type: none"> Portable detector integration Mobile Perception Go app that can be paired with PRDs Real-time detector readings displayed on UI Configurable alarm limits for each measurement type Locations of Perception Go app users and their detectors on a real-time map 			

<ol style="list-style-type: none"> 6. Customizable reports 7. Inventory tracking 8. User management 9. Media library - upload videos and photos from the scene 10. Information sharing - share data with trusted partner organizations in a few clicks
Available plugins (number of plugins limited by the subscription type)
<ul style="list-style-type: none"> • Live camera feed • Task management • UAV integration • Stationary device integration • Vehicle integration • Historical data playback • Device maintenance scheduling and tracking • Drill and exercise functionality (simulations) • Hazard modelling (e.g. plume)

Figure 38. Pricing page proposal for Perception Cloud

Executive management reviewed the pricing proposal and made several suggestions:

- A free tier was suggested for a small number of users and educational organizations
- The prices could be increased more drastically between tiers
- The highest tier doesn't have to specify an exact price, it can be made "request quote only"
- Instead of allowing customers to pick their own plugins, plugins can be priced separately, and each tier can come with a predetermined set of plugins.
- The number of devices should be included as a value metric. In the current proposal, only the number of device types is restricted with each tier. Server resource consumption increases when more devices are connected to the system, due to increased traffic – which raises the costs for the vendor.

Based on this feedback, an updated pricing page was created. The updated pricing page is presented in Figure 39. Several decisions were made when designing the updated pricing page:

- The prices of individual plugins are not listed on this page. The page can be enhanced later to better communicate that they can enhance their tier with selected plugins without upgrading to the next tier. On the one hand, that makes pricing more

complex, but on the other, this may be able to attract customers that are unwilling to pay the full price for the next tier.

- The price for each tier is currently not consumption-based. It could be possible to change the fixed monthly price to the price per user per month, for example. More market research is needed to determine whether this would be acceptable to potential customers. In the absence of this research, the product manager opted for the simpler version of pricing where each tier is priced at a fixed level.
- A header was added to the pricing page that imitates a link to a separate pricing page for the industrial edition of Perception Cloud. This will be absent from the initial version of the product, but it was discussed that industrial organizations are a desirable customer segment to target in the future.

Are you an industrial organization? View Perception Cloud Industrial Edition			
Basic	Responder	Defender	Premium
Free	700€ /month (billed annually) OR 800€ month-to-month	2500€ /month (billed annually)	Need more devices or features, or a more flexible plan? CONTACT US
No training fee	Required 1600€ training fee	Required 4000€ training fee	
<ul style="list-style-type: none"> ✓ Up to 10 user accounts ✓ Up to 15 detectors of 3 different detector types 	<ul style="list-style-type: none"> ✓ Up to 50 user accounts ✓ Up to 50 detectors of 15 detector types 	<ul style="list-style-type: none"> ✓ Unlimited user accounts ✓ Unlimited detectors of all supported detector types ✓ All features included ✓ Possible dedicated server for your organization 	<ul style="list-style-type: none"> ✓ We can integrate your device to be supported by Perception Cloud ✓ We can implement custom features to fit your needs
TRY	TRY BUY Free 30-day trial	TRY BUY Free 30-day trial	CONTACT US

<ul style="list-style-type: none"> ✓ Portable detector integration ✓ Mobile Perception Go app that can be paired with PRDs ✓ Real-time detector readings displayed on UI ✓ Configurable alarm limits for each measurement type ✓ Locations of Perception Go app users and their detectors on a real-time map ✓ User management 	<p>All Basic features, plus:</p> <ul style="list-style-type: none"> ✓ Live camera feed integration ✓ Customizable reports ✓ Inventory tracking ✓ Media library - upload videos and photos from the scene ✓ Information sharing - share data with trusted partner organizations in a few clicks 	<p>All Responder features, plus:</p> <ul style="list-style-type: none"> ✓ Drill and exercise functionality (simulations) ✓ Hazard modelling (e.g. plume) ✓ Task management ✓ UAV integration ✓ Stationary device integration ✓ Vehicle integration ✓ Historical data playback ✓ Device maintenance scheduling and tracking 	<p>Custom plan and features to fit your unique needs</p>
--	--	---	--

Figure 39. Updated pricing page proposal for Perception Cloud

This pricing page was added to the product strategy document for Perception Cloud. The design decisions listed above were also added to the document for context.

5.2.5. Legal review

The legal review consists of checking four legal aspects: contracts, IPR protection, open source, and data protection.



Figure 40. Functional areas of commissioner company involved in legal review

Figure 40 shows the functional areas of the commissioner company whose representatives participated in the phase.

Data protection discussion

The data protection discussion of the legal review phase took place on July 11th, 2022. The meeting was held virtually. The commissioner company has a dedicated Data Protection Officer, and the product manager held a one-on-one discussion with the Data Protection Officer. In preparation for the meeting, the product manager read the data protection page, which the commissioner company has on the company intranet. The page outlines the core data protection principles in the organization and links to relevant templates, process models, and current policies for existing software services. To facilitate efficient discussion, the product manager prepared the meeting agenda and made a short slide deck to guide the discussion. The contents of the slide deck were as follows:

1. Agenda
 - a. Brief introduction of Perception Cloud strategy so far
 - b. Data protection considerations
 - c. Contract considerations
2. Strategy so far
 - a. Link to the current product strategy document
3. Data protection considerations
 - a. Do we need a new privacy policy for Perception Cloud? Can we reuse existing policies? It looks like the company policy was updated in 2022, so it is up to date
 - b. Should we require the users to accept the privacy policy both on the web and mobile?
 - c. Should we require the users to accept the cookie policy on the web and mobile?
 - d. What regional issues can we run into? E.g. if the customer is in [Country X] or [Region Y]?
4. Contract considerations
 - a. Outside the responsibilities of the Data Protection Officer, so we can be brief here
 - b. Are you aware of any service license agreements that we already have for other SaaS products?

At the beginning of the meeting, the product manager introduced the product strategy so far – the problem and position statement from the collaborative strategy workshop, the prioritized list of features from the product definition refinement phase, the domain and

business process models from the architecture design phase, and the draft pricing page from the financial discussion phase. The introduction of architecture and pricing was high-level. The product manager focused on aspects that may have an impact on data protection activities. For example, when architecture was presented, the focus was on data sharing with partner agencies – a single event project being shared by multiple organizations.

Based on the features and architecture, the Data Protection Officer pointed out the identifiable user location data as high-risk personal information in the system. Perception Cloud stores operator locations linked to their user accounts. Event projects can be replayed using the “history playback” feature, meaning that the location is stored in the system.

During the discussion, “data controller” and “data processor” were the key terms. The commissioner company has a company-wide privacy policy that defines its responsibilities as a data controller. The commissioner company has a separate privacy policy for a different SaaS product X. For SaaS product X, the commissioner company is the data processor. The organizations using product X are data controllers for their employee data. The commissioner company is the data controller for its own customers, meaning it manages administrator accounts for each of the organizations. The privacy policy for product X links to the company-wide policy to define the role of the commissioner company as the data controller of administrator accounts. To reiterate, the commissioner company is the data controller for its own customer data, but the data processor for the customer organization data.

It was agreed that the commissioner company can largely reuse the privacy policy for product X for Perception Cloud. According to GDPR, a contract needs to be made between the data controller and a data processor when a new customer signs up for Perception Cloud. This contract is called the Data Processing Agreement (DPA) in GDPR terminology. This contract can be added as an appendix to the Perception Cloud SLA. The commissioner company already has a template for this contract. The template is only available in Finnish and needs to be translated. An example of the G Suite Agreement was given. The main agreement acts as the SLA, and links to the DPA. The G Suite Agreement and the Data Processing Amendment can be found at the following links:

- G Suite Agreement:

https://admin.google.com/terms/apps/1/9/en/premier_terms_eea.html

- Data Processing Amendment: https://workspace.google.com/terms/dpa_terms.html

Next, it was discussed whether Perception Cloud should require the user to accept the privacy policy both on web and mobile UI. It was agreed that this isn't necessary. The privacy policy of the commissioner company has several clauses explaining when it applies. The legal bases for data processing are presented in Figure 41. While "consent" is one option, "performance of a contract" is another. When the customer accepts the SLA, the commissioner company can process the personal data as part of performing the contract made between the commissioner company and the customer.

Commissioner company processes the personal data only when at least one of the following legal bases for processing applies:

- a. **Consent:** a person or a user has given consent to the processing of his or her personal data for one or more specific processing purposes.
- b. **Performance of a contract:** Processing is necessary for the performance of a contract to which the person or the user is party or in order to take steps at the request of the person or the user prior to entering into a contract.
- c. **Legal obligation:** processing is necessary for compliance with a legal obligation to which the commissioner company is subject.
- d. **Protection of vital interests:** processing is necessary in order to protect the vital interests of the person or the user or of another natural person.
- e. **Public interest or official authority:** processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the commissioner company
- f. **Legitimate interest:** processing is necessary for the purposes of the legitimate interests pursued by the commissioner company or by a third-party.

Figure 41. Legal bases for personal data processing by the commissioner company. These are all available legal bases in GDPR. In order to be lawful, data processing needs to comply with one of these bases

Regarding the cookie policy, it was determined that according to the current plan, the software will only use cookies for essential operations, and not to identify and track the users. Because of this, an explicit cookie acceptance popup is not required at this point.

Next, possible regional issues were discussed. It was agreed that Perception Cloud would comply with GDPR. While GDPR compliance is not mandatory for customers outside of EEA, managing a separate data protection approach for each country would make the process convoluted. GDPR is comprehensive and would apply to all customers by default, regardless of their country of origin. Additional data protection steps can be negotiated if requested by the customer.

Free trials were discussed separately, and it was agreed that all customer data would be deleted after some time if they do not wish to continue after the free trial.

Finally, the contracts were briefly discussed. The COO has been in charge of formulating SLAs historically. The product manager would discuss the SLA with the COO. The contract between the data processor and the data controller must be placed as an appendix to the SLA.

As the result of the discussion, the Data Protection Officer proposed several actions to be taken next:

1. Map and document all types of personal information processed in Perception Cloud
2. Create Privacy Policy where the commissioner company is in the data processor role (the Product X Privacy Policy can largely be reused)
3. Describe the mapped data processed, how it is processed, data retention, sub-processors and data transfers outside of EEA
4. Create a Data Processing Agreement (DPA) which is an agreement between a data controller (the customer organization of the commissioner company) and a data processor (the commissioner company). The Finnish template was obtained from the Docue online legal compliance service. An English template can be obtained from there as well.
5. DPA can be added as an attachment to the Service License Agreement (SLA) for Perception Cloud
6. The agreement should be legally binding, either signed physically or digitally

The key takeaways and next steps were added to the product strategy document.

Open source check-up

The open source check-up was carried out independently by the product manager on July 23rd, 2022. The two major open source components of Perception Cloud are Keycloak and Grafana. The product manager checked the terms of their licenses. Keycloak is distributed under Apache License 2.0 (<https://github.com/keycloak/keycloak/blob/main/LICENSE.txt>), and Grafana is distributed under AGPLv3 (<https://grafana.com/licensing/>). Both of these allow distribution for commercial use. AGPLv3 is a copyleft license that poses a risk of having to make the source code public when misunderstood. The license requires making the source code public if the software vendor modifies the source code of the open source component. In the case of Perception Cloud, the pre-built distributable of Grafana is used without modifications. Reporting dashboards are configured using the standard features of Grafana, without modifying the source code.

Contract discussion & IPR protection check-up

The contract and IPR protection discussions were conducted together in one meeting, which took place at the office of the commissioner company on July 29th, 2022. The COO discussed these aspects with the product manager. Regarding the contracts, the commissioner company is using the IT2022 service (<https://it-ehdot.fi/>), which provides a collection of general and special terms and conditions for companies selling IT services and products. The terms and templates are sold by a division of the Finnish Chamber of Commerce and are designed to help small and middle-size companies create legal documents (IT2022, 2022).

The participants also browsed through the contract of an existing SaaS product sold by the company. That contract was written several years ago using a custom format. It was decided to use the new IT2022 template instead of adapting this existing contract. In particular, the “IT2022 ETP Special terms and conditions for services delivered via data network (cloud service)” contract template was selected.

Regarding IPR protection, it was agreed that the commissioner company should purchase trademarks for the product. But obtaining patents was deemed unnecessary at this point. Obtaining a patent comes at a significant cost, which increases with the geographical scope of the patent. Both trademarks and patents can be purchased via the Finnish Patent and Registration Office (<https://www.prh.fi/en/index.html>).

The product manager made list of actions to be taken next:

1. List the brand names that need to be trademarked, and apply for trademarks for these brands.
2. Fill the blank IT2022 template “IT2022 ETP Special terms and conditions for services delivered via data network (cloud service)”. Refer to the ready-made terms in the template, and override them where deemed appropriate.
3. Add the DPA as an appendix to the terms and conditions.

This list was added to the product strategy document.

5.2.6. KPI selection & risk analysis

The KPI selection was conducted on July 24th, 2022. The product manager used the Balanced Product Scorecard, as well as the list of sample KPIs suggested by Pichler (2016).

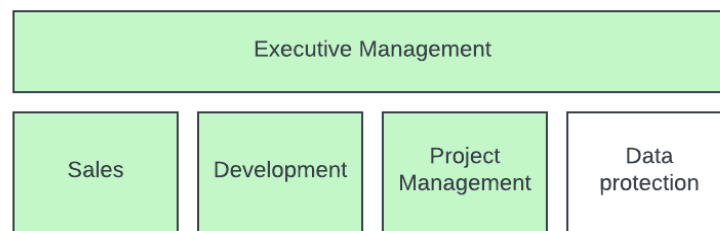


Figure 42. Functional areas of commissioner company involved in the KPI selection & risk analysis

Figure 42 shows the functional areas of the commissioner company whose representatives participated in the phase. The KPIs selected by the product manager are presented in Table 9.

Table 9. KPIs selected for Perception Cloud

Category	Name	Way to measure	Business goal target
Business goals	New revenue source	Calculate revenue generated by the product one year after release. Calculate the total revenue of the company in the year after the product launch. Calculate the percentage of total revenue brought by the new product.	The product should increase company revenue by 5 to 10% within one year after launch. (Line in

			the sand metric to measure progress)
	Reduced cost of customer acquisition	Calculate the average cost of acquisition for the customer-specific version of the software. Compare it to the cost of acquisition calculated for the new product.	It should become cheaper to obtain new customers and configure the software for them. The cost of customer acquisition should decrease by 30% within one year after launch.
Financial KPIs	Revenue	Accounting data, the total amount paid by each customer	
	Cost	Accounting data, cost of developing the product and making releases	
	Cost of acquisition	Accounting data, the total amount spent on marketing materials, conducting demos to customers, travelling to customers to showcase the product, conducting installation and training.	
	Cost of hosting and support	For hosting - total bills by the selected cloud provider. If support is outsourced, use bills by the customer support organization. If support is done in-house, calculate the percentage of monthly hours each employee spent providing customer support. Multiply the monthly salary by that percentage to calculate how much customer support cost for each person.	
	Customer lifetime value	Monetization. Total revenue from each customer minus the cost of acquisition, support, and hosting	
Customer KPIs	Retention	The number of customers that signed up and did not cancel their subscriptions.	
	Cancellation rate	The number of customers that cancelled divided by the total number of customers	
	Conversion rate	The number of customers that continued beyond the trial period or upgraded from the free tier, divided by the total amount of customers that used the trial or signed up to the free tier	
	Complaints and support requests	The number of support requests, and the severity of each request	

	Customer and user feedback	Analyse feedback received from various channels. Is it positive, neutral, or negative? What features or aspects are praised or criticized?
Product and Process KPIs	Product quality	SonarQube server will be set up for the project. A quality gate will be enforced for the project using SonarQube. SonarQube will provide reports on code quality, code smells, coverage, bugs and possible security issues.
	User interaction	Integrate activity monitoring into the application. Record which are the most and least used features. Must research available software tools for this - developing this from scratch may be costly.
People KPIs	Team knowledge and skills	Talking to the team during sprint retrospectives and monitoring project progress, development speed and the product quality KPI.

The selected KPIs were added to the product strategy document and sent to the stakeholders from executive management, Development and Sales. The participants were asked to comment on the product strategy overall, including the KPIs. The participants were also requested to mark three statements or elements of the strategy that they are least confident about.

The product strategy document was reviewed and approved by the selected stakeholders. The KPI selection was also approved, but an important issue was raised regarding the selected targets for the business goals. Increasing annual revenue by 5 to 10% was deemed unrealistic. At the moment, the annual revenue of the company is tied to the number of contracts obtained for professional software services, and the delivery of these services. Because of this, the revenue varies each year. The goal for the new SaaS is to provide a steady and growing source of revenue, but comparing it to overall company revenue will not provide a meaningful measure of success. A long-term average of the annual Perception revenue was suggested as one possible point of comparison, but no decision was made. Selecting the proper target for the financial success of the product thus requires further consideration.

The exclusion of SMS integration and chat functionality – a decision that was made during product definition refinement – raised questions with a stakeholder that did not participate in that phase. It was also mentioned that the free tier in the draft pricing page needs to be

further simplified. It currently offers up to 10 accounts and up to 15 detectors – that should be changed to 3 accounts and 3 detectors. The stakeholders wrote that the purpose of the free tier is to be a very limited teaser of the product, additionally serving as a way to collect contacts of customers that are willing to try it.

Several areas of the strategy were pointed out as risky. At this point, it was not possible to prioritize the list of risks in terms of their severity, and the risks are simply identified. The identified risks are as follows:

1. Selection of the cloud provider. Many of the organizations in our target market have regional and security restrictions on where the cloud services that they use are allowed to be hosted. Adopting any of the major cloud providers (AWS, Google, or Azure) can be undesirable by some of the customers. The selection of the cloud provider for Perception Cloud still needs to be carefully considered.
2. The decision to prioritize real-time response features over post-factum analysis features was seen as a source of risk. A representative from executive management wrote that real-time response features can be seen as the bare minimum for such a system. Perception Cloud may lose customers to the competition offering robust post-factum capabilities. With the proliferation of machine learning and Big Data, Perception Cloud may need to enhance its data analysis capabilities to meet the increasing demand for these technologies, especially when the competition in the market increases.
3. The technical risk of developing and maintaining two separate codebases for Perception Cloud and customer-tailored Perception was discussed. The company intends to keep developing an on-premises version of Perception. The company needs to plan the versioning of the software, the process to create releases from the shared code base, the development of necessary CI and developer tooling, and ensuring the developers understand the implications of their changes to the shared code to all the versions of Perception.

The “Digital Red Dot” game thus allowed to identify the possible risks, but at this point did not allow ranking the risks in order of severity. Risk analysis, risk planning, and risk monitoring activities still need to be established for a more rigorous risk management approach.

6. Model evaluation

This chapter provides a summative evaluation of the artefact, i.e., an evaluation conducted after the design is completed. To evaluate the process model, the thesis needs to determine how well the artefact solved the explicated problem and fulfilled the defined requirements (Johannesson and Perjons, 2021). Figure 43 displays the current stage in the design research process.

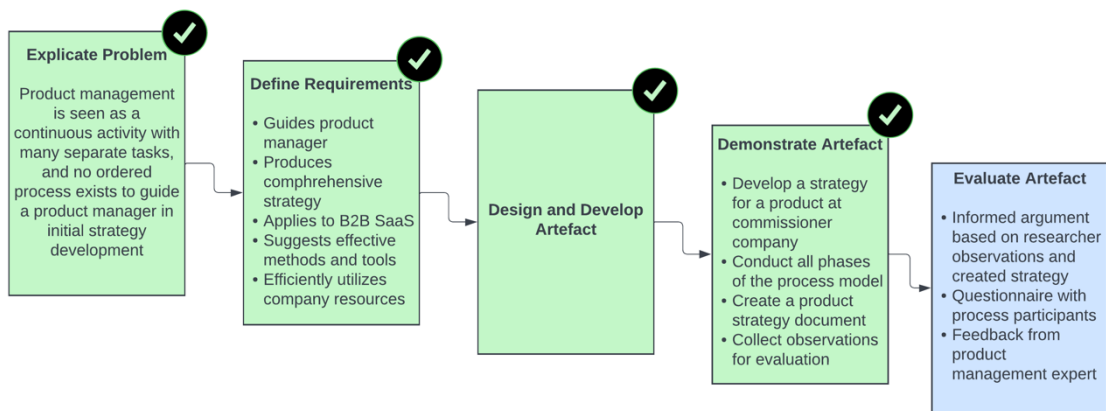


Figure 43. Design research process after demonstrating the artefact

The problem was explicated and the requirements were defined in Chapter 3. To reiterate, the problem was formulated as follows:

“Product management is seen as a continuous activity with many separate tasks, and no ordered process exists to guide a product manager in initial strategy development”.

The following requirements were defined for the artefact produced in this thesis – a process model for initial product strategy development for B2B SaaS:

1. The model shall provide direction to a product manager in charge of establishing a strategy.
2. The model shall lead to the creation of a product strategy.
3. The resulting strategy shall describe all 8 elements of the “Product strategy” column of the ISPMA SPM framework.
4. The resulting strategy shall be applicable to a B2B SaaS software product.

5. The model shall suggest methods and tools that are effective for B2B SaaS strategy development.
6. The model shall utilize company resources efficiently by ensuring only the necessary stakeholders are required to attend certain phases.
7. The resulting strategy shall be documented in a single product strategy document, which can be used to communicate the strategy across the organization.

The evaluation strategy for this thesis is naturalistic, meaning “the artefact is evaluated in the real world, within the practice for which it is intended (Johannesson and Perjons, 2021)”. The artefact in question is a business process model for initial strategy development, which supports productization of customer specific software into B2B SaaS. The author deemed evaluating such a model in an artificial setting not useful and not feasible. The process model was employed at the commissioner company during the demonstration described above. It was not possible to fully separate the demonstration phase from the evaluation phase within the timeframe of this work. An initial product strategy for the product in question can only be developed once, and the author of this work did not have access or time to develop a strategy for a second product, possibly at a different company.

Demonstration is mentioned by Johannesson and Perjons (2021) as a weak form of evaluation, suitable to present the artefact in a convincing and vivid way. The author observed the participants and collected notes while conducting the phases of the process model. Based on the observations collected and the resulting strategy, the author makes an informed argument on the success of the process model overall, and the fulfilment of the stated requirements. To complement the evaluation by observation, a questionnaire was conducted with the participants of the process. Johannesson and Perjons (2021) write that a questionnaire is an efficient method to collect the perceptions and opinions of many stakeholders, even though the responses can be superficial and prevent deep insights into the views of the respondents. For an even further removed independent evaluation, the draft of the thesis work was shared with Hans-Bernd Kittlaus – a founding board member and the current chairman of ISPMA, and a renowned product management expert, who over the course of his career held the position of the Head of Software Product Management and Development units at IBM, amongst other achievements. The informed argument based on

observation, the results of the questionnaire, and the criticism provided by Hans-Bernd Kittlaus are presented below.

6.1. Evaluation by observation

As for requirement 1, the process model did guide the product manager in the creation of the product strategy. But in this case the product manager was also the author of the process model. This introduces subjectivity into the evaluation. To further evaluate how well the model guides product managers in strategy creation, the model needs to be tested by a different product manager independently. The process model was designed with the commissioner company in mind. Because of this, there is a risk of “over-fitting” the model to the processes and the organizational structure of the thesis commissioner company. The thesis therefore cannot strongly claim that the model is universally applicable. An independent implementation of the process can test whether the model is generalizable and illustrate how the model can be generalized further.

With regard to requirement 2, the proposed process model was able to produce an initial product strategy for the commissioner company. The strategy was reviewed and approved by company stakeholders, creating alignment around the vision and priorities for the product. The strategy was able to focus and develop the productization ideas that were floated by various company stakeholders over the years but remained elusive until now. The resulting strategy is initial, with multiple strategy elements containing lists of actions to be taken next.

As for requirement 3, while the strategy covered all the elements of product strategy as classified by the ISPMA SPM framework, certain aspects were less refined. In particular, financial management was addressed scarcely. The Financial Management element was not defined beyond the business model – the revenue sources and cost factors. In fact, the “Financial discussion” phase of the process model only covered the “Pricing” knowledge area, omitting for example the creation of the business plan, or considering cost management. In its current form, the “Financial discussion” phase can be renamed into “Pricing discussion”. The phase can also be expanded with additional steps to address the missing financial elements. Ebert and Brinkkemper (2014) observe that product managers that come from a technical background prioritize the areas of the product strategy that they understand the best – the specification of requirements and the architecture design. The author of this thesis comes from a software engineering background and adopted the product

manager position for this thesis research. Looking at the product strategy document produced at the end of the strategy development process, indeed the architecture and requirements areas are defined in more detail than the financial management and KPI selection. The thesis thus supports the observation by Ebert and Brinkkemper.

To address requirements 4 and 5, the recommendations in the process model were based on the academic and practitioner literature aimed at B2B SaaS. The process model was able to produce a product strategy for the B2B SaaS product Perception Cloud. Most of the steps and suggested tools are also applicable to licensed software products and hybrid models – the Perception Cloud product itself includes a mobile app component. This implies that the process model could have broader applicability, beyond B2B SaaS. Pricing and legal aspects, however, are tied to the SaaS nature of the product. The Accion pricing framework used during the phase is designed for SaaS products, and the advice is less applicable to other product types, such as licensed software. The legal discussion was focused on the contract contents specific to SaaS (such as the SLA), and the data protection concepts applicable for delivering a service over a network (such as the data controller and the data processor). Elaboration of the financial discussion and legal review phases can make the process model more applicable to products other than SaaS. The tools and processes followed during the 6 phases of the process model elicited insightful feedback on the strategy, and the participating stakeholders were interested in discussions and enthusiastic to offer ideas.

The Accion framework – the tool selected for the pricing discussion – helped quickly explain the SaaS pricing concepts and best practices to the stakeholders that were not used to managing pricing decisions. The tool helped select the value metrics and the price structure for the new product. The participants, however, were unable to determine an upper bound that can be charged from customers. The thesis calls for further research in the area of price setting in the security and defence software markets. The BSC tool alone resulted in a non-systematic KPI selection process (selecting from sample KPIs guided by identified best practices), and the thesis recommends a further study of rigorous step-by-step KPI selection methodologies such as the “winning KPI methodology” by Parmenter. Nonetheless, both the Accion framework and the BSC produced a useful starting point for further strategy work, and illustrated which areas are well-understood, and which must be developed further.

Considering requirement 6, the selection of attendees for each phase was not systematic. Here the work experience of the author played a role. Over the course of several years at the

company, the author has developed an understanding of whose input is relevant for each phase. The process model can be complemented by a generic stakeholder selection approach. For example, the Power-Interest Grid can be employed to determine the most influential participants (Eden and Ackermann, cited by Pichler, 2016). RACI matrices can also be developed to map the responsibilities of the available stakeholders. The recommendations can also be adjusted depending on the size of the organization. For example, for the “Legal review” phase, a dedicated legal department may only be available in large organizations.

The meetings and discussions during the phase were efficient, from the perspective of the product manager. Overall, only several hours were spent in meetings, with six stakeholders joining some of the sessions. In the next section, the perspective of the other meeting participants will be described.

Finally, requirement 7 was filled. The strategy was defined in a single document, which was shared with stakeholders for feedback. The process model does not enforce a strict template for this document, allowing each product manager to define it in a way that best fits their company. An example table of contents was provided in Figure 6.

6.2. Evaluation by questionnaire

A questionnaire was created and shared with participants of the strategy development process via Google Forms. Each participant was asked which phase they participated in, and then prompted to strongly agree, agree, remain neutral, disagree, or strongly disagree with the following statements:

1. My understanding of the Perception Cloud product improved after participating in the sessions
2. I felt like the sessions I attended were too long
3. I felt like I had nothing to contribute to the sessions
4. My expertise and contributions during the sessions influenced the results of the sessions
5. I feel more confident in the success of the Perception Cloud product after attending the sessions

The plural form was used for the word “sessions” in the statements, even though some respondents only participated in one session. The questionnaire ended with an optional open feedback section for the sessions that the participant attended.

All the 6 company stakeholders that participated in the strategy development process responded to the questionnaire. The results were encouraging. All respondents agreed or strongly agreed that their understanding of the product improved after participating in the sessions (4 agree, 2 strongly agree). All respondents felt that their expertise and contributions influenced the results of the sessions (100% agree), and they feel more confident in the success of the product after attending the sessions (5 agree, 1 strongly agrees). None of the participants felt that the sessions they attended were too long (4 disagree, 2 strongly disagree), nor felt like they had nothing to contribute (5 disagree, 1 strongly disagrees).

In the open feedback section, the process model was commended as “good work”, even “excellent work”, and a “well arranged, well thought and professional take on the process”. One respondent was glad that a strategy for the product was finally created: “This has been much-needed clarification of Perception product strategy for SaaS service provision”. Another respondent praised the “well prepared sessions with clear agenda”. Finally, one of the respondents liked the way the materials were prepared, and the way the product manager had a vision on how to move the discussion forward, especially given the fact that most participants did not prepare before attending the meetings.

The results show that the participants feel that they understood the product and the strategy better after participating in the sessions, contributed with their expertise, and their time was not wasted. This helps prove that requirement 6 was implemented successfully, and the company resources were utilized efficiently. Moreover, the participants that chose to share open feedback indicate they are happy with the resulting strategy.

6.3. Feedback by Hans-Bernd Kittlaus

Hans-Bernd Kittlaus criticized the generalizability of the model in his feedback. He wrote that the model is based on a waterfall approach, where activities are executed in a sequence. This may work for the thesis commissioner company, which already has knowledge and experience in the target market where it sells the on-premise version of Perception. Using that experience, it is possible to document the strategy following the process model.

However, for brand new SaaS products, this is not the case. A company that is creating new SaaS – and not productizing an on-premises system – cannot establish a strategy at the beginning of development, at least not fully. The company therefore follows a highly iterative approach, creating prototypes and MVPs while experimenting and learning. At this stage, there is a lot of back and forth, since the strategy elements are interconnected. The elements are revised again and again, until a consensus is reached. Therefore, Kittlaus wrote, the model is not universally applicable for new B2B SaaS product development. In its current form, the process model is linear and does not describe the conditions under which the phases need to be revisited. For example, a discovery made during legal review could force the company to revisit all the preceding phases. Conflicts like this are not addressed in the current version of the process model. Additionally, Kittlaus pointed out that the Perception product is rather unique. In his opinion, it is worth further evaluating to what extent the uniqueness of the product has affected the process model itself.

The main points of criticism were summarized by the author of this thesis as follows:

1. The linear waterfall approach of the process is not applicable to brand new B2B SaaS products, where a company has no idea what the product should be or what the market wants at the beginning of development.
2. The process model does not sufficiently address the interdependencies between the phases, and does not describe the steps to go back and revise certain elements of the strategy when new information becomes available, decisions are changed, or conflicts are identified.

In the next chapter, the thesis will discuss the implications of these concerns, the limitations of the model that they help highlight, and the possible future enhancements of the model with this feedback in mind.

7. Discussion

The goal of this thesis stated in the introduction was to **design a process that produces a comprehensive product strategy for a company undergoing productization of customer-specific software into B2B SaaS**. To help achieve this goal in an informed and rigorous way, three research questions were formulated. The questions will now be answered in turn.

- **RQ1:** What process can a company follow to produce a comprehensive initial product strategy?

The study employed a design research methodology by Johannesson and Perjons (2021) to answer this question. The proposed model was described in detail in Chapter 3, demonstrated at the commissioner company in Chapter 4, and evaluated in Chapter 5. The model is grounded in the established SPMBOK, and consists of six phases: collaborative strategy workshop, product definition refinement, architecture design, financial discussion, legal review, and KPI selection & risk analysis. During each phase, one or several elements of product strategy as defined by the ISPMA SPM framework is discussed and refined. The results of the discussions are written down into a product strategy document, which is evaluated by key stakeholders at the company.

The process model designed in this thesis aspires to contribute both to theory and practice by providing prescriptive knowledge on how a company can create an initial product strategy for a B2B SaaS product based on their previously customer-specific software. The process model is centred around the comprehensive list of knowledge areas of product strategy as defined by the ISPMA SPM framework – the underlying knowledge area of the SPMBOK.

The study was motivated by a real business problem. The commissioner company needed to productize an existing software system into a standard software product. The goal was to offer the product to CBRN responder organizations as a SaaS. For that new product, a strategy was required. The commissioner company had no established product management practices. The study set out to determine the state-of-the-art practices in the discipline of SPM, and the knowledge areas that need to be covered by a product strategy to be considered holistic and complete. To tackle the business problem, the study developed a process model

for initial product strategy development, which supports productization of customer-specific software into B2B SaaS. The study took the design science research approach to create a process model that aspires to provide prescriptive knowledge for organizations and SPM practitioners. The design science research process calls for developing novel artefacts that solve “practical problems of general interest (Johannesson and Perjons, 2021)”. Solving the business problem for the commissioner company alone would fail to meet the “general interest” aspect of this definition. With that in mind, the study aimed not only to create a process that works for the thesis commissioner company, but could also have practical applicability for other companies productizing their customer-specific software into B2B SaaS.

The evaluation showed the model to be an efficient tool for product strategy development within the productization context of the thesis commissioner company. It resulted in a detailed product strategy that created alignment between multiple company stakeholders regarding the direction for the new product. The participants of the process felt their time and expertise were used efficiently, and the company approved the resulting product strategy document. The process helped structure and focus the productization ideas that were brought up by company stakeholders over the years, but remained elusive until now.

However, a review by an SPM expert revealed limitations in the applicability of the process model for brand new B2B SaaS product development outside of the productization context. In the case of the commissioner company, there was an initial understanding the market need addressed by the new product, and a general idea of the features that the product will offer. This helped follow the strategy development in a linear fashion. Brand new product development calls for iterative approaches, involving extensive learning, prototyping, and going back and forth. The generalizability of the model for brand new B2B SaaS development was indeed not demonstrated and not proven in this thesis. But the research goal stated in the introduction was to “**design a process that produces a comprehensive product strategy for a company undergoing productization of customer-specific software into B2B SaaS**”. The thesis did find the model effective for that goal at least in the case of the commissioner company, and calls for further testing of the model in that context at other organization. Still, even in the context of productization, the model can be enhanced by emphasizing the need for iteration. One new possible competitor for the product was identified after the product definition refinement phase was completed, and competing

factors were already defined in a Strategy Canvas. Situations like this, when new information becomes available, should act as triggers to revisit certain phases of the strategy development process.

The process model may act as one of the tools that helps introduce Software Product Management practices in an organization. Productization occurs at professional software service companies that recognize a shared market need across multiple projects. Such companies may be unaware of the state-of-the-art SPM practices of the SPMBOK. As Ebert and Brinkkemper (2014) write in their industry evaluation of SPM, companies piloting product management approaches must be supported by templates, examples, and training materials. The model proposed in this study could act as one such template, and the demonstration of the model could serve as an example of its application.

Especially during the product definition refinement phase, it can be useful for a product manager to have some experience in the product or market domain. This helps evaluate competing products and existing software to create a list of competing factors for the industry value curve. This is not a requirement, however, and most of the phases can be executed without deep domain experience. Incidentally, it would be natural to expect that a manager assigned to a product in a certain domain has some relevant experience in the area, or the means to acquire it.

The process model suggested and demonstrated the usefulness of several tools recommended by SPM and software business practitioners, including the Product Vision Board, the Strategy Canvas, the Accion Pricing Framework, the Balanced Product Scorecard, and the “Red Dot Game”. The thesis also suggested software engineering and business modelling tools - the domain model and BPMN business process models – to be used in the context of SPM. Testing the applicability of practitioner tools within the SPMBOK context allows incorporating new tools into this growing body of knowledge if they prove to be efficient. The demonstrated usefulness of these tools in an academic context contributes towards the development of a trusted SPM toolkit.

- **RQ2:** How does the B2B SaaS context influence product strategy decisions?

The main influence of the B2B SaaS context is on the delivery model, tailorability architecture, pricing, legal, and performance management aspects of the product strategy. The impact on the delivery model is self-evident, since SaaS is a specific delivery model that

requires the product to be offered on demand over a network, supporting scalability and multitenancy. The B2B SaaS context calls for the tailorability architecture to incorporate configurability and composability of the software. In B2B area, customization to specific business needs is often required by customers, but the multi-tenant SaaS model does not allow to freely customize the software for one customer without impacting others. Companies customizing their SaaS software products for customers that request it embark on a dangerous route that might negate the benefits of the SaaS delivery model and lead to isolated codebases for each customer. Focusing on a composable architecture and giving the customer configuration options to personalize their experience is the recommended approach for B2B SaaS vendors.

SaaS pricing is a complex area of research, with some researchers (see Saltan and Smolander, 2019) calling for a separate SaaS Pricing Body of Knowledge to be created in order to address the gaps between existing academic research and plentiful widely variable suggestions from the industry. Saltan and Smolander (2019) identified 13 pricing frameworks that offer various recommendations. Regardless of the specific framework, the pricing model for SaaS products is subscription-based – companies pay for the right to use the software at regular intervals. The subscription fee incorporates all or most product-related services, including maintenance, customer support, and data storage.

The pricing approach recommended for all types of software, including B2B SaaS, is value-based, not market-based or cost-based (Kittlaus and Fricker, 2017). In certain cases (e.g., when entering a mature market, or aiming to undercut the price of the competitors), it is still necessary to understand the price offered by competitors. In the B2B SaaS area, competitor pricing can be difficult to acquire to use as a baseline. The few identified competitor products do not specify their prices via pricing pages, but rather request customers to contact their sales departments.

In the area of legal management, the B2B SaaS context determines the type of contract offered to customers, and the data protection measures that need to be taken. A SaaS contract can be signed with individual customers, but is often offered in the form of standard terms and conditions. The contract includes a service level agreement, which may clarify the functional scope, the availability commitment, the backup policies, and the liability of the vendor should the terms be breached. Advanced data protection regulation such as GDPR also imposes restrictions on B2B SaaS software vendors, who host the server infrastructure

where personal data may be stored and processed. In GDPR terms, the B2B SaaS vendor can make a Data Processing Agreement with customers, which describes the types of data processed and the legal bases for processing the data. As countries worldwide follow in the footsteps of GDPR, all B2B SaaS vendors must understand their regional data protection regulations.

In the performance management area, the process model recommends a balanced approach to KPI selection – considering the financial, customer, product and process, and people perspectives. Still, the majority of the KPIs selected in the case of the commissioner company are focused on revenue, customer lifetime value, and monitoring customer activity to ensure retention and prevent churn, which is another influence of the B2B SaaS context.

- **RQ3:** How does the productization context influence product strategy decisions?

The productization context influences the product strategy decisions at least in positioning and product definition, pricing, sourcing, and financial management. In positioning, it is important to communicate to the market where the product stands in relation to the non-productized version of the system, especially if such a system continues to be sold in parallel with the standard product. In product definition, the selection of product features and attributes is to a significant extent predetermined by what is already offered in the non-productized system. The desire to reuse as much as possible of the existing codebase has been explicitly stated during the strategy design activities. The very decision to productize the system is driven by the recognition of generalized market needs and a standardizable product core within the existing customer-specific systems.

Selecting the price for the productized offering is a challenge. The sales team of a professional software services company may be used to receiving the price from customers as part of requests for proposals. Prices vary based on the project, and selecting a price independently proved to be challenging at the commissioner company. To address this lack of organizational experience in product pricing, the Accion pricing framework was used. Saltan and Smolander (2019) call for a practical evaluation of several action-oriented pricing frameworks, including Accion, and the thesis tested this framework at the commissioner company. Overall, the Accion framework was effective in creating an understanding of the pricing concepts and tactics for stakeholders that are not used to managing pricing decisions. The provided examples were illustrative, and the action-oriented nature of the framework

structured the discussion well. But defining the upper bound proved difficult for the Perception Cloud product. The Perception Cloud system is a CBRN security system intended to protect and save people's lives and health, and a further look at the insurance industry and the pricing of security systems could reveal established pricing approaches in this domain.

The organization productizing customer-specific software into standardized SaaS must be prepared to allocate resources to the long-term maintenance and support teams required for such products. The business models of standard software vendors and professional service providers differ. If both coexist in an organization, it is recommended to keep the professional service division separate from the standard product division. The implications of productization on the business model of the overall software organization were not considered in detail in this thesis, but it is part of financial management to determine the business model for individual products.

Having discussed the answers to the research questions, the limitations of the study can now be considered. A limitation pointed out by a product management expert who reviewed the model is its waterfall-like linear structure. Such a structure is not representative of a new product development process, where rapid experimentation and learning require a company to continuously revisit various elements of a product strategy in no particular order. This thesis only shows the process was applicable in the productization context at the commissioner company, and calls for further testing at other companies and with other kinds of products to make further conclusions about its generalizability and illustrate how it can be enhanced to be more generally applicable.

Notably, the thesis allows for the process model to be used and employed at different levels of abstraction. Figure 19 displays the model as a linear process, and shows which elements of the product strategy are impacted at each phase. Table 5 illustrates the model in a different way, specifying the key questions that need to be answered during each phase, and the tools that may help in answering these questions. Finally, Section 4.2 provides step-by-step instructions to conduct each phase. If, for example, the step-by-step instructions prove to be overly restrictive for a company trying to use the model, they could revert to the key questions that need to be answered during the phase, and disregard the steps in Section 4.2, or define their own.

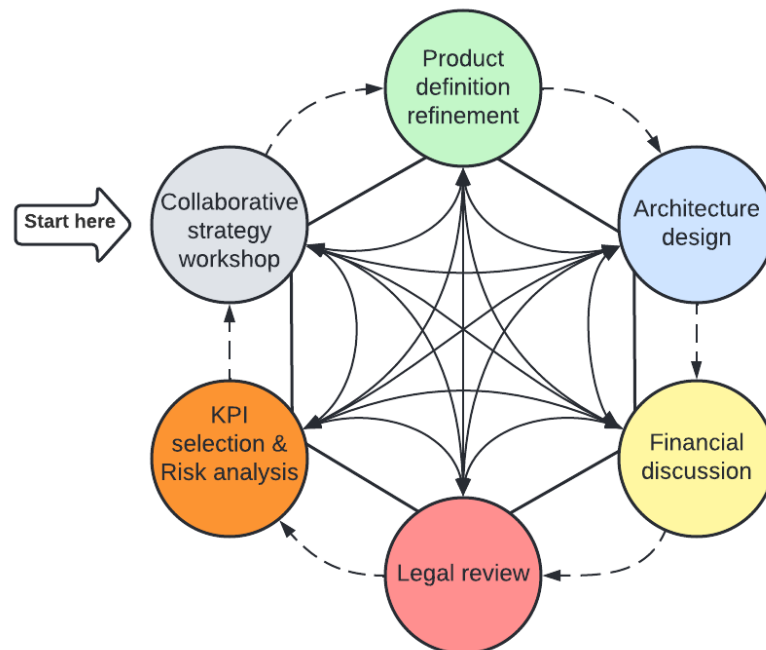


Figure 44. Possible new perspective on the model that emphasizes iteration

To address the limitation of the linear structure of the model, a variation in Figure 44 could be evaluated. This variation allows to freely move between the phases in any order. The bubbles representing each phase of the process model are arranged in a circle, with arrows connecting each phase to every other phase. The model still indicates that the process can be started from the collaborative strategy workshop, but after that is possible to jump to any other phase as needed. The linear order of the original version of the model is hinted at by connecting the bubbles with dashed arrows around the perimeter of the circle, but that order is only preserved as a general guideline and is non-restrictive. This variation was proposed by the author based on the results of the evaluation, and has not been tested during this thesis work. The variation can be enhanced with a more detailed description explicitly recommending to revisit certain phases when conflicts in the strategy are identified, or any of the strategy elements need further refinement.

Another limitation of the process model is that it does not explicitly involve talking to potential buyers or end users of the software. Discussing needs with the customers and creating buyer and user persona profiles is an efficient method to understand the value that the product should bring to these customers (Pichler, 2016). This was another consequence of the productization context, where the needs of the customer segment were already largely known to the sales team and the management of the commissioner company. Moreover, in

preparation for the product definition refinement phase, the product manager was able to find a recorded training session for one of the competing products, where end customers of that product demonstrated how they use it, shared what problems it is solving for them, and tips how to use it most efficiently. Still, the process model can be enhanced with an additional phase where potential customers are interviewed about their needs, or a focus group is arranged, for example. This could be a part of the product definition refinement phase, or a separate phase that precedes it.

One of the stated artefact goals was that the resulting strategy comprehensively covers all 8 elements of the product strategy in the SPM framework classification. The resulting strategy covers some of the elements only at a limited capacity. In particular, financial management is limited to the fundamentals of the business model – listing the revenue sources and cost factors. This was partly due to the vast scope of the product strategy knowledge areas and the limited timeframe of the research. Another reason may lie in the engineering background of the product manager and the author of the thesis in this case. As Ebert and Brinkkemper (2014) explain, many product managers follow this career trajectory and focus on areas of product strategy that they understand the best, such as requirement specification and architecture. In the strategy developed in this thesis, the product definition is indeed defined in more detail than financial management or KPI selection aspects.

No generic process was established in the process model for KPI selection and continuous performance management. Only the best practices for KPI selection and sample KPIs were provided. Investigation of certain action-oriented KPI methodologies is suggested in the directions for future research. Risk management was also limited to risk identification and prioritization. The model can be enhanced with ways to create risk avoidance and contingency plans, and set up procedures for continuous risk monitoring.

The process model is focused on the development of a strategy for a new product, which is a productized version of existing customer-tailored systems. But according to Ebert and Brinkkemper (2014), 80% of product management work is dealing with existing products, not new products and innovations. While the model was designed for creating a new standard product based on a customer-specific system, the described activities could be executed for existing products as well. Potentially, the model could be applied to improve an existing product strategy, making the state of the product more explicit, and identifying and addressing any blind spots in its strategy. Testing the applicability of the model for existing

product improvement, and adjusting or adding to its phases, is one possible direction for future research. Several other directions for future research were identified:

- Testing the model at a different company, by a product manager that is not the author of the process model. One of the artefact requirements for the process model is to provide guidance to a product manager in charge of crafting a strategy. As explained in Chapter 7, in the case of this thesis, the author of the process model and the product manager is the same person. Executing a process model independently from its author would provide a more objective perspective on how well it is able to guide a product manager.
- Cross-disciplinary research between scholars of software engineering, product management, and finance to simplify the selection of the upper bound, the maximum value that a SaaS vendor may charge from customers. As explained in the Demonstration and Evaluation Chapters, one of the challenges in pricing design for the product was the determination of this upper bound.
- Pricing of security systems. Security systems deal with the cost of life and health, and it could be interesting to look at the insurance industry and see if any practices from that area can be used to estimate the value of the system. Another similarity with insurance is that in security systems the incidents are typically rare. The security system brings the most of its value in the rare cases when incidents occur, but still needs to be set up and paid for continuously. Communicating value to customers must be done well to prevent churn.
- Complementing the process model with the roadmap development. As pointed out both by Kittlaus and Fricker (2017) and Bavaro and McDowell (2021), a product strategy document shall include a roadmap for the product. The roadmap creation is part of product planning – an area of product management responsibilities that was excluded from the scope of this thesis. Efficient tools and methods for roadmap development can be studied and documented next.
- Investigating the more systematic KPI selection methods, such as the “Winning KPI methodology” by Parmenter. As explained in the Demonstration and Evaluation Chapters, the Balanced Product Scorecard alone resulted in a KPI selection process that is not systematic. More rigorous and opinionated KPI selection approaches, such

as the one proposed by Parmenter, require further attention. For example, it is interesting to see how these more advanced processes can be implemented by small and medium-sized companies, which may not have the resources to create dedicated KPI teams, as Parmenter suggests.

- Developing a taxonomy of product management tools. This thesis selected several tools for the process model but did not aim to provide a comprehensive list of tools or a comparison between them. Further research in this area could produce an SPM toolkit. The process model could then be enhanced to recommend several options from the toolkit for each of the phases.

A new perspective on the process model emphasizing iteration was suggested above in Figure 44. A further development of this perspective could increase the generalizability of the model.

8. Conclusion

The novel part of this study is a process model for initial product strategy development for B2B SaaS that supports companies productizing their customer-specific software. The model consists of six phases: collaborative strategy workshop, product definition refinement, architecture design, financial discussion, legal review, and KPI selection & risk analysis. During each phase, one or several elements of product strategy as defined by the ISPMA SPM framework is discussed and refined. The process model was demonstrated at a Finnish B2B software company, which is productizing its customer-specific systems into B2B SaaS. The process successfully produced a product strategy and created a shared understanding and alignment between multiple stakeholders at the company.

The process model was evaluated by collecting observations during its demonstration, conducting a questionnaire with the stakeholders who participated in the process, and a review by a product management expert. In the case of the commissioner company, the model was found to guide the product manager in designing the product strategy, use the time and expertise of various stakeholders efficiently, suggest tools and processes that help elicit insightful feedback, and produce a product strategy that addresses all elements of a product strategy to an extent from limited to detailed. The study found that the B2B SaaS context of the product influenced product strategy in delivery model, tailorability architecture, pricing, legal, and performance management aspects. The productization context influenced the positioning and product definition, pricing, sourcing, and financial management elements of the strategy.

The implication of this thesis for SPM practitioners is that the process model may provide prescriptive knowledge for the development of initial product strategies for B2B SaaS, at least in the context of productization. The productization context implies that a company may have some knowledge of what the product should be and what the market is, based on experience of providing professional software services to that market. Product managers may use the process model to introduce product management activities in their organizations. Improving software product management activities at companies provides immense benefits for businesses. Time to market, staying within schedules, and product quality all sustainably improve with proper SPM activities in place (Ebert and Brinkkemper, 2014). Following the

process model, a product managers can produce a strategy that addresses in some capacity all the required knowledge areas related to product strategy. The model may prevent overlooking relevant aspects during strategy design.

The process model overall may contribute to the SPMBOK as a template to be used by companies piloting product management approaches, and the demonstration of the model could act as an example of its successful application. Ebert and Brinkkemper (2014) write that the implementation of product management practices must be supported by examples, templates, and training tailored to the business model and market positioning of the company (Ebert and Brinkkemper, 2014, p.17). The thesis proposed the usage of certain tools suggested by SPM practitioners operating outside of the theoretical framework of the SPMBOK. The demonstration of such tools in this thesis provides a data point that can lead to the inclusion of these tools in the SPMBOK, and the creation of a comprehensive product strategy development toolkit.

The linear structure of the model was criticized as a limitation preventing its use for new product development outside of the productization context. In cases when the product and the market are completely unknown at the beginning of the process, it is necessary to iterate and experiment, rapidly moving back and forth across the process, and executing strategy design activities in different order in an ad hoc fashion. Still, the goal stated in the introduction of this thesis was to **design a process that produces a comprehensive product strategy for a company undergoing productization of customer-specific software into B2B SaaS**. The thesis was able to achieve that goal at the commissioner company and calls for further testing of the model at other companies undergoing productization. The initial product strategy for Perception Cloud was adopted by the commissioner company and will guide product development. Various elements of the strategy will be revisited when more information becomes available, and certain decisions change. One new possible competitor has already been identified, and the product definition refinement will be conducted again to evaluate how it affects the competing factors. The thesis also suggests a possible future modification of the model that emphasizes iteration to make it more applicable for brand new product development outside of the productization context.

Another limitation of the process is that it addresses some of the elements of product strategy to a lesser extent than others. In particular, the area of Financial Management is limited to the revenue sources and cost factors of the business model. A business plan or cost

management aspects are not yet addressed. The study calls for further development and testing of the process model, and continuous refinement of its phases. In the sphere of pricing and financial management, a cross-disciplinary study with scholars of management and finance can further advance the process model. A comprehensive toolkit for product strategy development can also be created. The process model should be applied and evaluated at other companies, and by product managers other than the author of the model.

Arnold Toynbee wrote that “no tool is omniscient (Toynbee, 1964, cited by Epstein, 2019)”. The process model proposed in this thesis was aimed to solve a real business problem for the thesis commissioner company. In the spirit of design science, which creates artifacts to solve practical problems of general interest, the thesis aspired to offer generalizable instructions to product managers in charge of productization efforts. In designing the model, the author strived to achieve balance between offering advice that is too vague and creating a process that is too tied to a specific software company. While the evaluation of the process model at other software companies is yet to be performed, the results of its first application were encouraging. The resulting product strategy created alignment at the commissioner company, and the model itself was commended by the executive management as a well-thought-out, well-arranged, and professional take on the process. Future testing and development of the model can further fine-tune its phases, expand its focus to licensed and embedded software, and address other areas of product management, including product planning and orchestration. The development of a new variation of the model emphasizing iteration could make it applicable for more cases, including product development outside of the productization context.

References

Accion (2015). *Pricing your SaaS product*. [online] www.accion.org. Available at: <https://content.accion.org/wp-content/uploads/2018/08/Pricing-Your-SaaS-Product.pdf> [Accessed 13 Jul. 2022].

Accion (2022). *What is Accion?* [online] www.accion.org. Available at: <https://www.accion.org/about/what-is-accion> [Accessed 18 Jul. 2022].

Artz, P., van de Weerd, I., Brinkkemper, S. and Fieggen, J. (2010). Productization: Transforming from Developing Customer-Specific Software to Product Software. *Lecture Notes in Business Information Processing*, pp.90–102. doi:10.1007/978-3-642-13633-7_8.

Bavaro, J. and McDowell, G.L. (2021). *Cracking The PM Career*. S.L.: CareerCup.

Bizagi (2022). *Free business process mapping and modeling software - Bizagi Modeler*. [online] www.bizagi.com. Available at: <https://www.bizagi.com/en/platform/modeler> [Accessed 20 Jul. 2022].

Burke, A., Van Stel, A. and Thurik, R. (2016). Testing the Validity of Blue Ocean Strategy versus Competitive Strategy: An Analysis of the Retail Industry. *International Review of Entrepreneurship*, 1529(14(2)), pp.123–146.

Buxmann, P., Diefenbach, H. and Hess, T. (2013). *The Software Industry*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Cambridge Dictionary (2019a). *POSITIONING* | *meaning in the Cambridge English Dictionary*. [online] Cambridge.org. Available at: <https://dictionary.cambridge.org/dictionary/english/positioning> [Accessed 10 Apr. 2022].

Cambridge Dictionary (2019b). *STRATEGY* | *Meaning in the Cambridge English Dictionary*. [online] Cambridge.org. Available at: <https://dictionary.cambridge.org/dictionary/english/strategy>.

Cambridge Dictionary (2022). *Business-to-business*. [online] Cambridge.org. Available at: <https://dictionary.cambridge.org/dictionary/english/business-to-business> [Accessed 6 Aug. 2022].

Carton, G. (2021). The Story Behind the Business Model Canvas. *Entrepreneur & Innovation Exchange*. [online] Available at: <https://eiexchange.com/content/the-story-behind-the-business-model-canvas> [Accessed 28 Jul. 2022].

CBRNResponder (2021). *CBRNResponder Nationwide Introductory Training*. [online] [www.youtube.com](https://www.youtube.com/watch?v=1a1VqoB-LWY&t=950s&ab_channel=CBRNResponder). Available at: https://www.youtube.com/watch?v=1a1VqoB-LWY&t=950s&ab_channel=CBRNResponder [Accessed 24 May 2022].

CBRNResponder (2022). *CBRNResponder - CBRNResponder Network*. [online] www.cbrnresponder.net. Available at: <https://www.cbrnresponder.net/> [Accessed 7 Aug. 2022].

Dempsey, D. and Kelliher, F. (2017). *Business-to-Business Client Relationships in the Cloud Computing Software as a Service Realm. Industry Trends in Cloud Computing*, pp.83–109. doi:10.1007/978-3-319-63994-9_5.

Doshi, S. (2021). *Choosing, refining, and tracking product metrics*. [online] Mixpanel. Available at: <https://mixpanel.com/blog/shreyas-doshi-product-metrics/> [Accessed 13 Jul. 2022].

Ebert, C. and Brinkkemper, S. (2014). Software product management – An industry evaluation. *Journal of Systems and Software*, 95, pp.10–18. doi:10.1016/j.jss.2013.12.042.

Epstein, D. (2019). *Range: Why Generalists Triumph in a Specialized World*. New York: Riverhead Books.

European Commission (2019). *What is a data controller or a data processor?* [online] European Commission. Available at: https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/obligations/controller-processor/what-data-controller-or-data-processor_en [Accessed 21 Jul. 2022].

Fowler, M. (2015). *Patterns of enterprise application architecture*. Boston, Mass. ; Munich: Addison-Wesley.

Gartner (2012). *Definition of Multitenancy - Gartner Information Technology Glossary*. [online] Gartner. Available at: <https://www.gartner.com/en/information-technology/glossary/multitenancy>.

Hevner, A., March, S., Park, J. and Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, [online] 28(1), p.75. doi:10.2307/25148625.

Honeywell (2022a). *ConneXt Gas Monitoring*. [online] sps.honeywell.com. Available at: <https://sps.honeywell.com/us/en/services/safety/gas-detection-management/connext-gas-monitoring> [Accessed 7 Aug. 2022].

Honeywell (2022b). *Safety Suite*. [online] sps.honeywell.com. Available at: <https://sps.honeywell.com/us/en/software/safety/connected-safety/safety-suite> [Accessed 24 May 2022].

ISPMA (2020). *SPM Summit: Customer-Specific Tailorability of Your B2B Software Product with Hans-Bernd Kittlaus*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=p8hO0rxENTl&ab_channel=ISPMA [Accessed 1 Aug. 2022].

ISPMA (2021a). *Framework – ISPMA*. [online] Available at: <https://ispma.org/framework/> [Accessed 10 Apr. 2022].

ISPMA (2021b). *Software Product Management Foundation Level Syllabus. V.2.0. Student Edition*. [online] Available at: <https://ispma.org/wp-content/uploads/2021/04/ISPMA-SPM-EL-Syllabus-ProductStrategy-V.2.0-Student-Edition.pdf> [Accessed 10 Apr. 2022].

IT2022 (2022). *Briefly in English*. [online] it-ehdot.fi. Available at: <https://it-ehdot.fi/briefly-in-english/> [Accessed 29 Jul. 2022].

Johannesson, P. and Perjons, E. (2021). *Introduction To Design Science*. S.L.: Springer Nature.

Kaplan, R.S. and Norton, D.P. (1996). Linking the Balanced Scorecard to Strategy. *California Management Review*, [online] 39(1), pp.53–79. doi:10.2307/41165876.

Kim, W.C. and Mauborgne, R. (2005). *Blue Ocean Strategy: How to Create Uncontested Market Space and Make the Competition Irrelevant*. Boston, MA: Harvard Business School Press.

Kim, W.C. and Mauborgne, R. (2013a). *Blue Ocean Strategy & Blue Ocean Shift Tools and Frameworks*. [online] Blue Ocean Strategy. Available at: <https://www.blueoceanstrategy.com/tools/>.

Kim, W.C. and Mauborgne, R. (2013b). *Eliminate-Reduce-Raise-Create Grid (ERRC Grid) | Blue Ocean Strategy Tools and Frameworks*. [online] Blue Ocean Strategy. Available at: <https://www.blueoceanstrategy.com/tools/errc-grid/>.

Kim, W.C. and Mauborgne, R. (2013c). *Strategy Canvas*. [online] Blue Ocean Strategy. Available at: <https://www.blueoceanstrategy.com/tools/strategy-canvas/> [Accessed 21 Jul. 2022].

Kittlaus, H.-B. (2022). *Software Product Management. The ISPMA®-Compliant Study Guide and Handbook*. 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-662-65116-2.

Kittlaus, H.-B. and Fricker, S.A. (2017). *Software Product Management*. Berlin, Heidelberg: Springer.

Koch, R. (2019). *Cookies, the GDPR, and the ePrivacy Directive*. [online] GDPR.eu. Available at: <https://gdpr.eu/cookies/?cn-reloaded=1> [Accessed 21 Jul. 2022].

Lucid Software (2017). *UML Class Diagram Tutorial*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=UI6lqHOVHic&ab_channel=LucidSoftware [Accessed 20 Jul. 2022].

OMG (2011). *Business Process Model and Notation (BPMN)*. [online] <https://www.omg.org/>. Object Management Group, Inc. Available at: <https://www.omg.org/spec/BPMN/2.0/PDF> [Accessed 20 Jul. 2022].

OMG (2022). *View the BPMN Quick Guide*. [online] BPMN Quick Guide. Available at: <https://www.bpmnquickguide.com/view-bpmn-quick-guide/>.

Paajoki, A. (2020). Best practices for and benefits from implementing ISPMA's SPM framework. *jyx.jyu.fi*. [online] Available at: <http://urn.fi/URN:NBN:fi:jyu-202009145848> [Accessed 24 May 2022].

Parmenter, D. (2015). *Key performance indicators : developing, implementing, and using winning KPIs*. 3rd ed. Hoboken, N.J.: Wiley.

Pichler, R. (2016). *Strategize: Product Strategy and Product Roadmap Practices for the Digital Age*. Pichler Consulting.

Product School (2019a). *How to Have the Best B2B Approach in Product by Google PM*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=0syFtxbkaoQ&ab_channel=ProductSchool [Accessed 18 Mar. 2022].

Product School (2019b). *How to Transition to B2C vs. B2B Product Management by Amazon PM*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=GSsHJT1-NW4&ab_channel=ProductSchool [Accessed 18 Mar. 2022].

PWC (2013). The future of software pricing excellence: SaaS pricing. [online] Available at: <https://www.pwc.com/mt/en/publications> [Accessed 18 Jul. 2022].

Rantanen, J. (2022). [Forthcoming]. Transferring a situational awareness system to a cloud environment for use by several organizations. [online] Available at: <https://lutpub.lut.fi/>.

Salameh, A. and Bass, J. (2019). Spotify Tailoring for B2B Product Development. *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. doi:10.1109/seaa.2019.00018.

Saltan, A. (2021). *Revealing the State of Software-as-a-Service Pricing*. [online] *lutpub.lut.fi*. Lappeenranta-Lahti University of Technology LUT. Available at: <http://urn.fi/URN:ISBN:978-952-335-724-2> [Accessed 10 May 2022].

Saltan, A. and Smolander, K. (2019). Towards a SaaS Pricing Cookbook: A Multi-vocal Literature Review. *Lecture Notes in Business Information Processing*, pp.114–129. doi:10.1007/978-3-030-33742-1_10.

Saltan, A. and Smolander, K. (2021). How SaaS Companies Price Their Products: Insights from an Industry Study. *Lecture Notes in Business Information Processing*, 407, pp.1–13. doi:10.1007/978-3-030-67292-8_1.

Sommerville, I. (2011). *Software engineering*. 9th ed. Boston: Pearson, Cop.

Visual Paradigm (2019). *UML Class Diagram Tutorial*. [online] Visual-paradigm.com. Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/> [Accessed 20 Jul. 2022].

Visual Paradigm (2022). *Online Strategy Canvas Tool*. [online] online.visual-paradigm.com. Available at: <https://online.visual-paradigm.com/diagrams/features/strategy-canvas-software/> [Accessed 20 Jul. 2022].

Wolford, B. (2018). *What Is GDPR, the EU's New Data Protection law?* [online] GDPR.eu. Available at: <https://gdpr.eu/what-is-gdpr/> [Accessed 21 Jul. 2022].

Yrjönkoski, T. (2018). *How to support transformation from on-premise products to SaaS?: Position paper for future research*. [online] researchportal.tuni.fi. Available at: <https://researchportal.tuni.fi/en/publications/how-to-support-transformation-from-on-premise-products-to-saas-po> [Accessed 23 May 2022].

Appendix 1. Sample Key Performance Indicators by Roman Pichler (2016)

Perspective	Sample KPIs	Brief Description
Financial	Revenue	How much revenue is your product generating?
	Cost	What is the cost of developing and launching major releases or product versions?
	Cost of acquisition	How much does it cost to acquire a customer?
	Profit	How much profit is the product making? Note that you may want to track different profit types, including net and gross profit.
	Customer lifetime value	How much profit do individual customers create across the entire future relationship?
	Cash flow	Is the cash flow positive or negative? If negative, when do you expect to reach the break-even point?
Customer	Market share	How big is your market share compared to the competition?
	Adoption rate	Is your product gaining traction in the marketplace? If so, how quickly?
	Engagement	How engaged are the users? How many active daily users does the product have?
	Retention	How many customers are coming back?
	Net promoter score and referrals	How likely is it that people will recommend the product? How many people are actually recommending it?
	Cancellation rate	How many contracts are cancelled?
	Complaints and support requests	How many customer complaints and support queries do you receive? How severe are they?
	Conversion rate	How well are inquiries and evaluations translated into sales?
	Customer and user feedback	Do the customers and users have a positive, neutral, or negative attitude toward your product? What reviews are they providing, what feedback do they submit, and what do they tell you when you talk to them?
Product and Process	User interaction	What are the most and least common user journeys? Where do most drop-offs occur? What are the most and least used features?
	Product quality	Is it easy to change and extend the product? How high are the code complexity and the refactoring potential? How high is the test coverage? How many bugs are found and closed?
	Development process	How effective is the development process? Does it support the work of the team? Does the team have the right environment and the right tools?
	Schedule variances	Do new product versions and major releases meet their objectives? Are they deployed on time and on budget?
People	Team motivation	Is the development team motivated to work on the product? How high are turnover rates and absenteeism?
	Team knowledge and skills	Does the team have the necessary knowledge to do a good job? Does it improve its skills and acquire new knowledge?
	Stakeholder engagement	Do the stakeholders regularly participate in strategy and roadmap reviews and in sprint review meetings?
	Management sponsorship	Do you have the right management sponsor? Is the sponsor interested in the product and its performance?